



UNICEUB – CENTRO UNIVERSITÁRIO DE BRASÍLIA
FATECS – FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

EDUARDO MOREIRA BRAGA NETO

CONTROLE DE LUZ SOLAR EM AMBIENTES
POR POLARIZAÇÃO

BRASÍLIA / DF
1º SEMESTRE DE 2009

EDUARDO MOREIRA BRAGA NETO

**CONTROLE DE LUZ SOLAR EM AMBIENTES
POR POLARIZAÇÃO**

Monografia apresentada ao Curso de Engenharia da Computação, como requisito parcial para obtenção do grau de Engenheiro de Computação.

Orientador: Prof. Gil Renato Ribeiro Gonçalves

BRASÍLIA / DF
1º SEMESTRE 2009

Dedico esta monografia aos meus pais e amigos que tanto me apoiaram de forma direta ou indireta para que obtivesse êxito na conquista de mais uma vitória nesta longa jornada.

Agradecimentos

Agradeço principalmente a minha família e em especial meus pais Maurinda Maria Matos da Silva e José Eustáquio da Silva por me darem a oportunidade das conquistas alcançadas.

Agradeço ao meu irmão Bruno Moreira Matos que compartilha comigo momentos de luta na conquista deste mesmo objetivo me apoiando e ajudando um ao outro nas dificuldades encontradas.

Meus colegas de curso (principalmente Adriel Lar, Daniel Neto, Danilo Resende, Felipe Kladi, Luis Gustavo e Hugo de Souza, Ronie) que sempre presentes me ajudaram de forma direta ou indiretamente para alcançar mais este objetivo dando o devido apoio nos inúmeros momentos difíceis encontrados durante todo o curso e pelos inesquecíveis momentos de alegria proporcionados.

Não podendo deixar de citar o importante apoio de parentes como meu primo Diogo Pinto Portela e seu pai José Pinto de Matos.

Agradeço também ao corpo docente do UniCEUB pela paciência e disposição nas longas aulas ministradas no período noturno, onde todos se encontravam cansados depois de um longo dia de trabalho. Em especial agradeço ao professor Gil Renato pela ótima orientação e interesse ao meu projeto e à professora Maria Marony por importantes informações durante o processo de montagem do hardware.

Resumo

Este projeto implementa uma forma de controle de luz solar para ambientes de forma gradativa utilizando a polarização da luz. O protótipo do projeto possui duas películas polarizadoras que são as responsáveis por efetuar esta variação do nível de transparência do dispositivo. O projeto disponibiliza níveis de transparência pré-determinados para a escolha do usuário (100%, 75%, 50%, 25% e 0% de transparência), onde o nível inicial (100% de transparência) é dado pela transparência natural da película utilizada (em torno de 90% de transparência). Para o funcionamento do sistema é utilizado um motor de passo do tipo híbrido e bipolar é está controlado por um microcontrolador modelo AT89S52, que faz parte da família 8051, fabricado pela Atmel que também possui a função de receber a informação do usuário. O *software* que integra o microcontrolador é composto pelas linguagens C e *Assembly*

Palavras chave: Películas polarizadoras, microcontrolador e motor de passo.

Abstract

This project implements a way to control the gradual entry of sunlight into rooms using the polarization of light. The prototype project has two polarized films that are responsible for making this change in the level of transparency of the device. The project provided levels of transparency for the user choice (100%, 75%, 50%, 25% and 0% of transparency), where the initial level (100% of transparency) is given by the natural transparency of the used film (around 90% of transparency). For the operation of the system, it is used a step-type hybrid and bipolar which is controlled by a microcontroller AT89S52 model, which is part of the family 8051, manufactured by Atmel that also has the function of receiving the information of the user. The software that integrates the microcontroller is compounded by the C language and Assembly.

Keywords: Polarizing films, microcontroller and step motor.

Sumário

1	INTRODUÇÃO.....	14
1.1	Contextualização do Trabalho	14
1.2	Motivação	15
1.3	Objetivo do Projeto	15
1.4	Estrutura do Trabalho	16
2	INTRODUÇÃO TEÓRICA E PRINCIPAIS COMPONENTES UTILIZADOS	18
2.1	Polarização	18
2.1.1	Definição de Polarização	18
2.1.2	Início da Polarização na Indústria.....	18
2.2	Motor de Passo.....	23
2.2.1	Motores Elétricos	23
2.2.2	Motores de Passo.....	24
2.2.3	Passos.....	25
2.2.4	Tipos de Motores de Passo	26
2.2.4.1	Motor de Passo por Relutância Variável	26
2.2.4.2	Motor de Passo de Imã Permanente	27
2.2.4.3	Motores de Passo Híbridos	28
2.2.5	Formas de Operação.....	29
2.2.5.1	Motores Unipolares	29
2.2.5.2	Motores Bipolares.....	31
2.2.6	Controle do Motor de Passo	34
2.3	Microcontrolador 8051 AT89S52	35
2.3.1	Pinagem do Microcontrolador 8051 AT89S52	38
2.4	L298N	40
2.4.1	Ponte H.....	42
2.5	Ferramentas Utilizadas	43
3	IMPLEMENTAÇÃO DO <i>HARDWARE</i> E DO <i>SOFTWARE</i>	44
3.1	Descrição do <i>Hardware</i>	44
3.1.1	Motor de Passo Utilizado.....	44
3.1.2	CI L298N	47
3.1.3	Ligações do Microcontrolador 8051	49

3.1.3.1	Esquema dos Dispositivos Emissores de Sinal	52
3.1.3.2	Gravação do Software no Microcontrolador AT89S52	54
3.1.4	Pinagens do <i>Display</i> LCD 16X2	56
3.1.5	<i>Hardware</i> Finalizado.....	60
3.2	Desenvolvimento do <i>Software</i>	61
3.2.1	Linguagens Utilizadas para a Criação do <i>Software</i>	61
3.2.2	Fluxograma.....	62
3.2.3	Funcionamento do <i>Software</i>	63
4	RESULTADOS OBTIDOS.....	73
4.1	Mensagens de Interação com o Usuário	73
4.2	Montagem da Maquete	78
4.2.1	Imagens da Maquete	80
5	CONCLUSÃO	87
	REFERÊNCIAS BIBLIOGRAFICAS E SITES CONSULTADOS	89
	APÊNDICE	92
	ANEXO I – DATASHEET DO CI L298N.....	101
	ANEXO II – DATASHEET DO MICROCONTROLADOR AT89S52	106
	ANEXO III – DATASHEET DO <i>DISPLAY</i> 16X2.....	113

Índice de Ilustrações

Figura 1-1 – Visão geral do projeto	16
Figura 2-1 - Hidrocarbonetos não orientados	19
Figura 2-2 - Hidrocarbonatos orientados (folha plástica esticada)	20
Figura 2-3 – Representação de uma onda elíptica.....	20
Figura 2-4 – Representação da filtragem da onda de luz pelo Polaróide.....	21
Figura 2-5 - Filtragem da onda de luz	22
Figura 2-6 - Demonstração dos Polaróides efetuando a polarização da luz	23
Figura 2-7 – Representação da diferença angular entre os passos do motor de passo.....	25
Figura 2-8 – Esquema de um motor de passo por relutância variável	27
Figura 2-9 – Esquema de um motor de passo de ímã permanente.....	28
Figura 2-10 – Esquema de um motor de passo híbrido	29
Figura 2-11 – Demonstração das ligações das bobinas de um motor de passo unipolar	30
Figura 2-12 – Esquema de um passo inteiro do motor de passo unipolar	30
Figura 2-13 – Esquema de meio passo do motor de passo unipolar	31
Figura 2-14 – Demonstração das ligações das bobinas de um motor de passo bipolar.	32
Figura 2-15 – Esquema de passo inteiro do motor de passo bipolar	33
Figura 2-16 – Esquema de meio passo do motor de passo bipolar	33
Figura 2-17 – Diagrama do microcontrolador 8051	37
Figura 2-18 – Pinagem microcontrolador 8051	38
Figura 2-19 – CI L298N.....	40
Figura 2-20 – Pinagens do CI L298N	41
Figura 2-21 – Diagrama de Blocos do CI L298N.....	41
Figura 2-22 – Esquema Ponte H.....	43
Figura 3-1 – Parte de trás do motor de passo utilizado.....	46
Figura 3-2 – Motor de passo fechado.....	46
Figura 3-3 – Motor de passo aberto	47
Figura 3-4 – Circuito de ligação entre CI L298N e motor de passo com acionamento por <i>push buttons</i>	49

Figura 3-5 – Esquema de ligação do Cristal de Frequência.....	52
Figura 3-6 – Esquema de ligação dos push buttons e emissores de sinal.....	52
Figura 3-7 – Esquema do sistema de emissão de sinal	53
Figura 3-8 – Representação da armação de sustentação da película móvel com os metálicos de contato	54
Figura 3-9 – Ligações entre microcontrolador e porta paralela para gravação do software.....	55
Figura 3-10 – Interface do software Grava211	56
Figura 3-11 – Imagem do <i>display</i> LCD 16X2	58
Figura 3-12 – Esquema de ligações do <i>hardware</i>	60
Figura 3-13 – <i>Hardware</i> finalizado	61
Figura 3-14 – Fluxograma do <i>software</i> desenvolvido.....	62
Figura 4-1 – Demonstração da tela de inicialização.....	73
Figura 4-2 – Sistema disponibilizando apenas opção para escurecer	74
Figura 4-3 – Sistema disponibilizando apenas opção para clarear	74
Figura 4-4 – Sistema disponibilizando duas opções de movimento	75
Figura 4-5 – Sistema informando que a janela está escurecendo e caminhando para 50% da transparência total.....	75
Figura 4-6 – Sistema informando que a janela está clareando e caminhando para 50% da transparência total.....	76
Figura 4-7 – Imagem mostrada ao pressionar <i>push button</i> indevido quando janela estiver com nível máximo de transparência.	77
Figura 4-8 – Imagem mostrada ao pressionar <i>push button</i> indevido quando janela estiver com nível mínimo de transparência.	77
Figura 4-9 – Esquema da Película fixa.....	79
Figura 4-10 – Esquema da Película móvel.....	79
Figura 4-11 – Armação para sustentação dos componentes vista pela frente.....	81
Figura 4-12 – Armação para sustentação dos componentes vista por trás.....	81
Figura 4-13 – Armação para sustentação dos componentes mostrando a montagem entre as placas	82
Figura 4-14 – Caixa para encaixe das armações de sustentação dos componentes.....	83
Figura 4-15 – Caixa com armações de sustentação acopladas.....	83

Figura 4-16 – Maquete pronta mostrando películas com nível máximo de transparência.....	84
Figura 4-17 – Maquete pronta mostrando películas com 75% de transparência	84
Figura 4-18 – Maquete pronta mostrando películas com 50% de transparência	85
Figura 4-19– Maquete pronta mostrando películas com 25% de transparência	85
Figura 4-20– Maquete pronta mostrando películas com 0% de transparência	86

Índice de Equações

Equação (2.1) – Lei de Malus.....	22
Equação (3-1) – Lei de Ohm	59
Equação (3-2) – Calculo da queda de tensão para display.....	59

Índice de Tabelas

Tabela 2-1 – Tabela passo inteiro 1	34
Tabela 2-2 – Tabela passo inteiro 2	34
Tabela 2-3 – Tabela meio passo	35
Tabela 3-1 – Esquema de passos utilizados no projeto	45
Tabela 3-2 – Especificações e siglas da pinagem do <i>display</i>	57
Tabela 3-3 – Comando utilizados para configuração do <i>display</i>	57
Tabela 4-1 – Relação entre nível de transparência e ângulo entre películas	80

1 INTRODUÇÃO

1.1 Contextualização do Trabalho

Em um mundo cada vez mais competitivo, globalizado e capitalista, a busca pelo conforto e comodidade aumentando a cada dia. O fato da grande maioria das famílias terem o casal como trabalhadores para sustentar a casa torna a vida cada vez mais estressante e cansativa. Estes sintomas causados pelo excesso de compromissos podem ser alguns dos motivos pela procura de itens que proporcionem um conforto maior à família.

Outro ponto importante é a busca pela saúde. Com a industrialização veio o aumento da poluição, gerada pelos gases emitidos pela maioria das máquinas movidas à combustão. Com o nível de poluição cada vez maior houve também o aumento de algumas doenças respiratórias.

Este projeto traz como característica principal o conforto e comodidade gerada aos usuários através do controle da incidência de luz solar nos ambientes, pois ele permite a escolha do nível de claridade no ambiente durante o dia e também o nível de privacidade.

Outro benefício que este projeto traz para quem o utiliza é a opção de descartar as cortinas, quando utilizadas com a finalidade de controlar a entrada de luz solar no ambiente como as do tipo *Black-Out*, pois estes tipos de objetos geralmente são produzidos em tecido e podem gerar o acúmulo excessivo de poeira, ácaros e outros microorganismos que causam prejuízos à saúde.

Este projeto pode proporcionar também a diminuição do consumo de energia elétrica, uma vez que com um nível menor de luz solar atingindo o ambiente ele fica menos quente, isso pode ajudar a diminuir a utilização de condicionadores de ar que possuem um consumo considerável de energia, ou de lâmpadas acesas, onde a luz do sol pode iluminar o ambiente com intensidade controlada.

Outro ponto positivo é o fato de não se conhecer um dispositivo com estas funções e que utiliza materiais semelhantes, desta forma podendo estar surgindo um produto novo e de grande utilidade para seus adeptos.

1.2 Motivação

Através dos conhecimentos adquiridos durante o curso, surgiu o interesse em implementar um produto inicialmente com fins acadêmicos, mas que pode ser útil em residências e empresas tendo em vista um melhoramento na qualidade de vida, segurança e possivelmente um melhor rendimento no trabalho.

1.3 Objetivo do Projeto

O principal objetivo deste projeto é a implementação de *hardware* e *software* em um dispositivo para o controle de luminosidade através do uso da polarização da luz.

Este dispositivo deve ser implantado em janelas circulares pelo fato de utilizar duas películas polarizadoras paralelas, onde uma delas é fixa e a outra gira formando uma diferença angular de até 90° entre elas para ocorrer o efeito da mudança do nível de transparência desejada.

Inicialmente existem neste projeto cinco níveis de transparência, pré-determinados, para que o usuário possa escolher: 0%, 25%, 50%, 75% e 100% de transparência (dentro do total permitido pelas películas polarizadoras). Futuramente para a utilização comercial, este projeto pode sofrer alterações oferecendo outras opções de escolha ao usuário.

Para este projeto é utilizado um microcontrolador AT89S52 da família 8051 para dispensar a utilização de um microcomputador conectado ao sistema durante seu funcionamento. Este microcontrolador também tem como finalidade, a diminuição do consumo de energia elétrica.

O usuário tem a opção de escolher entre as transparências pré-determinadas através de uma interface onde são disponibilizados dois “*push buttons*”: um aumenta a porcentagem de transparência e outro diminui. Ao acionar um dos botões o sistema automaticamente posiciona as películas a fim de disponibilizar a próxima porcentagem de transparência, de acordo com a opção escolhida pelo usuário. A figura 1-1 mostra como será o esquema.

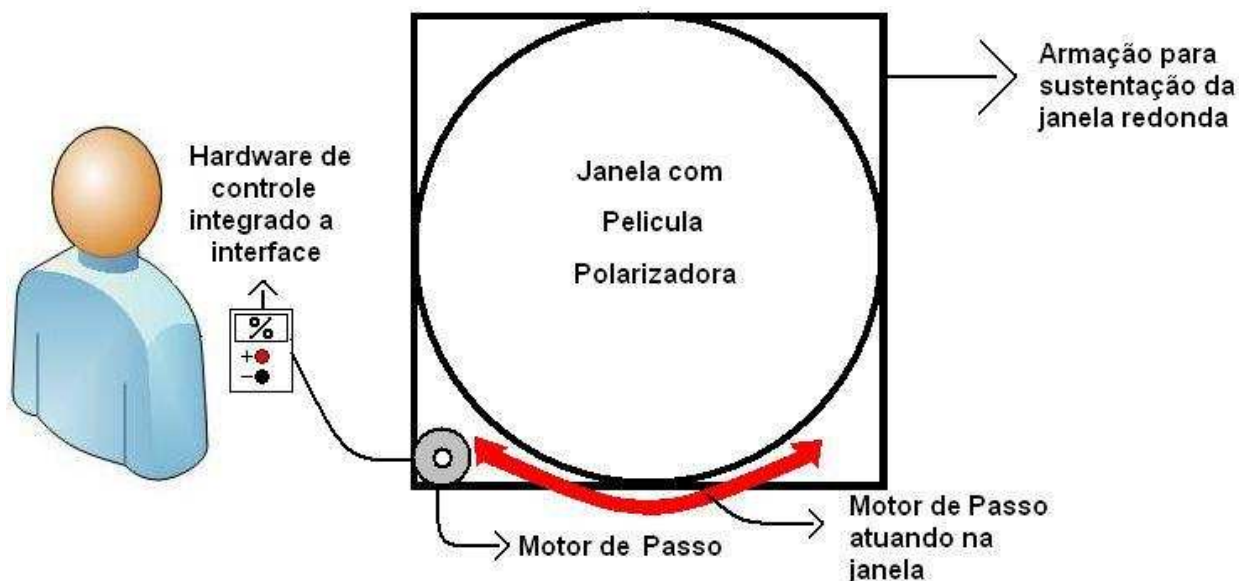


Figura 1-1 – Visão geral do projeto¹

1.4 Estrutura do Trabalho

Nesta monografia está disponibilizado além deste capítulo inicial (Introdução) os seguintes capítulos:

Capítulo 2 – Introdução teórica sobre os principais componentes utilizados. Este capítulo fala a respeito dos principais componentes utilizados explicando o seu funcionamento, não aprofundando no assunto já que se trata apenas de ferramentas;

Capítulo 3 – Implementação de *software* e *hardware*. Aqui é descrita a conexão entre os componentes do *hardware*, programação efetuada no microcontrolador e

¹ Fonte: Braga (2009)

utilização de alguns componentes eletrônicos necessários para funcionamento dos componentes principais ou somente para regulagem.

Capítulo 4 – Resultados obtidos. Aqui são informados os resultados obtidos na finalização do projeto, com demonstração por fotos da funcionalidade do sistema através da maquete criada.

2 INTRODUÇÃO TEÓRICA E PRINCIPAIS COMPONENTES UTILIZADOS

Este capítulo disponibiliza uma breve teoria dos principais componentes utilizados para a construção do projeto, com o objetivo de tornar mais fácil o entendimento do resultado final.

2.1 Polarização

2.1.1 Definição de Polarização

A luz emitida pela maioria das fontes de luz visível é despolarizada. Tal decorre do fato de a luz não ser exatamente monocromática. Para polarizar a luz usam-se nomeadamente polarizadores lineares, quer os do tipo polaróide, quer os feitos de material dicróico, (UC, 2007).

O polarizador linear mais simples consiste numa estrutura filamentar de condutores paralelos, separados de uma distância menor do que o comprimento de onda, \vec{E} . Os elétrons dessas estruturas condutoras absorvem o campo eletromagnético, oscilando na direção dos filamentos. Apenas a componente do campo na direção dos filamentos pode ser absorvida (os e^- não podem oscilar na outra direção). Por conseguinte, o conjunto absorve uma das componentes de polarização, mas deixa passar a outra. O eixo óptico do polarizador designa a direção da polarização que passa através dele, (UC, 2007).

2.1.2 Início da Polarização na Indústria

As aplicações industriais com a polarização tiveram início na década de 30 através do físico americano Edwin Land que conseguiu polarizar ondas de luz. Land foi um dos poucos cientistas que conquistaram lugar respeitado no mercado com seus produtos. Com a criação de um filtro polarizador para luz visível, barato e

eficiente, chamado de “Polaróide” Land inventou uma máquina fotográfica e fundou a Polaroid Corporation que ainda hoje é uma grande indústria, (UFC, 2008).

O filtro polarizador, produzido por Land em 1938 tinha característica de polarizar ondas de luz visíveis e sua produção era bastante simples, pois eram necessários basicamente uma folha plástica e uma solução que possuía o iodo como componente, (UFC, 2008).

O processo utilizado foi o seguinte: primeiro Land pegou uma folha de plástico, esta folha possuía longas moléculas de hidrocarboneto, que originalmente não possuem orientação, e a esticou fortemente em uma direção, este processo faz com que os hidrocarbonetos da folha de plástico passem do estado de desordem para o estado parcialmente alinhado. Após este processo, a folha esticada foi imersa na solução onde ocorreu uma reação que fez com que os átomos do iodo se ligassem às moléculas orientadas deixando-as eletricamente condutoras, retirado o material da solução e deixando-o secar, Land deixou a folha de plástico relaxar e percebeu que após este processo suas moléculas continuaram orientadas. As figuras 2-1 e 2-2 representam o processo sofrido pelas moléculas, (UFC, 2008).

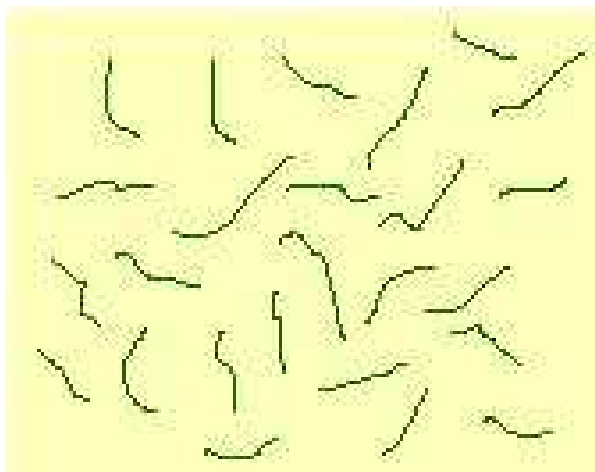


Figura 2-1 - Hidrocarbonetos não orientados²

² Fonte: Braga (2009)

Estes hidrocarbonetos ligados com os átomos de iodo fazem com que a folha de plástico funcione como um filtro para as ondas de luz, assim polarizando a onda, (UFC, 2008).

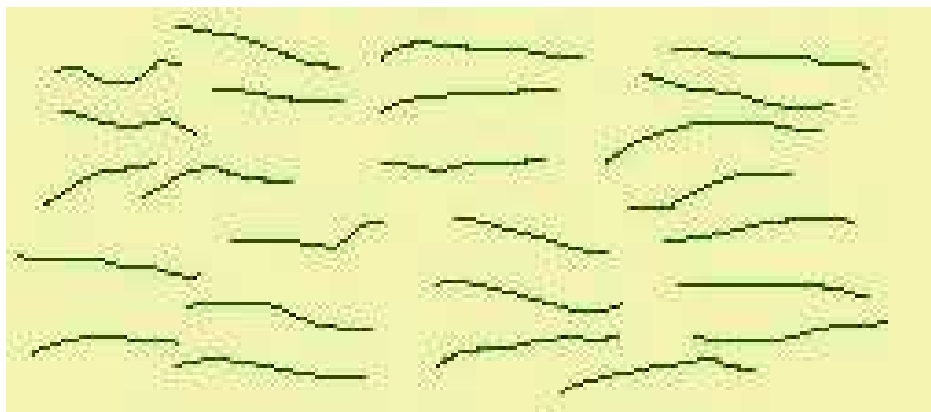


Figura 2-2 - Hidrocarbonatos orientados (folha plástica esticada)³

“A onda de luz é uma onda eletromagnética transversal (lado-a-lado) na qual campos elétricos e magnéticos oscilam perpendicularmente à direção de propagação da onda e a 90° com relação um ao outro.” (SERAFIN, 2008).

A onda de luz originalmente possui um formato elíptico, ou seja, ela se desloca efetuando movimentos próximos dos circulares conforme mostra a figura 2-3 (DA-RIN, 2005).

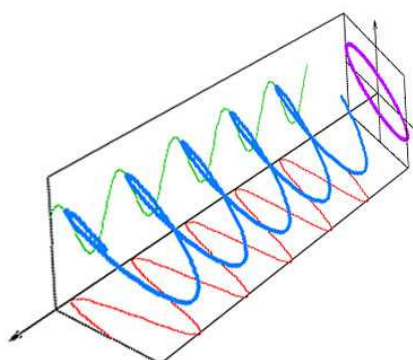


Figura 2-3 – Representação de uma onda elíptica⁴

³ Fonte: Braga (2009)

⁴ Fonte: <http://www.bpiropo.com.br/cf20050718.htm>. Acessado em 01 de março de 2009

A polarização é um fenômeno que pode ser caracterizado pela filtragem de uma onda eletromagnética, onde é incluída a onda de luz. A luz natural emitida pelo sol e a luz de lâmpadas comuns não estão sob o efeito deste fenômeno, ou seja, os campos eletromagnéticos de um raio de luz proveniente destas fontes estão vibrando em todas as direções perpendicularmente ao raio, formando assim a Elipse. O Polaróide possui a característica de filtrar estas ondas deixando passar somente as ondas que se encontram em um sentido, como mostra a figura 2-4, (UNICAMP, 2009).

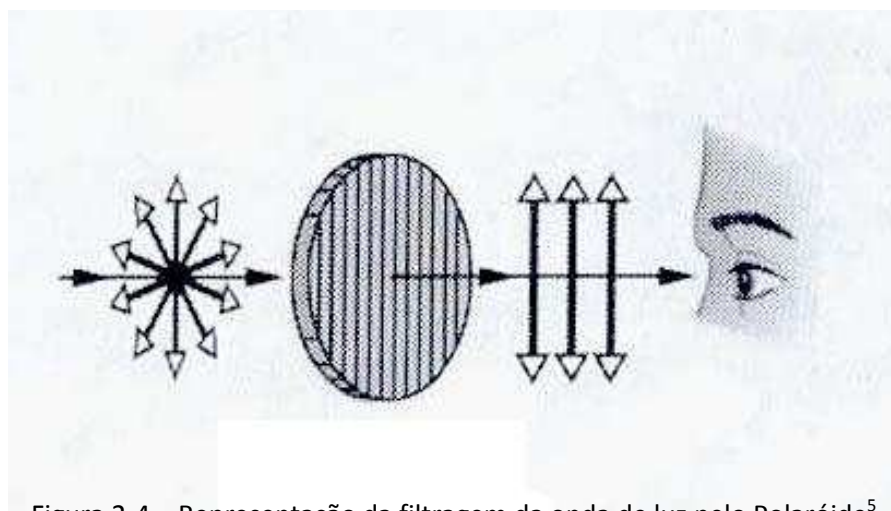


Figura 2-4 – Representação da filtragem da onda de luz pelo Polaróide⁵

Este fenômeno causado pelo polaróide é fundamental para o funcionamento do projeto aqui proposto que é acoplado a uma janela. Quando dois materiais polarizadores são colocados em paralelo e a angulação entre eles é alterada formando a diferença angular de até 90° , obtém-se uma filtragem da onda de luz em dois sentidos diferentes, assim quanto maior o ângulo formado pelos polaróides, no limite de 90° , menor será a intensidade da onda luminosa atravessando os polaróides. Como na figura 2-5, (UNICAMP, 2009).

⁵ Fonte: <http://www.ifi.unicamp.br/~accosta/roteiros/18/nota%2018.html>. Acessado: 25 de março de 2009.

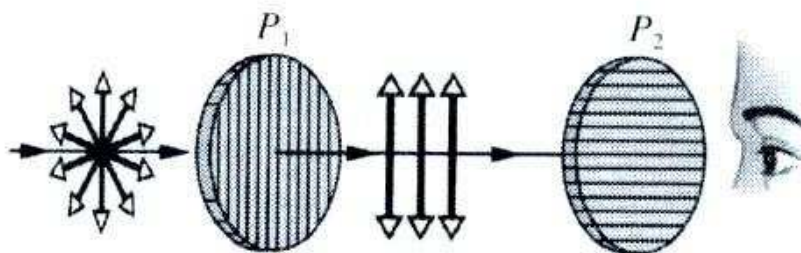


Figura 2-5 - Filtragem da onda de luz⁶

Esta filtragem ocorre através da absorção das ondas que se encontram no mesmo sentido dos hidrocarbonetos orientados, assim o polaróide permite passar apenas as ondas horizontais se estiver polarizando na vertical ou, as verticais se estiver polarizando na horizontal. Com a utilização de dois polaróides e efetuando a mudança de angulação entre o sentido da polarização, tem-se a polarização das ondas em dois sentidos fazendo com que um polarizador deixe, por exemplo, passar ondas somente na horizontal e outro na vertical, desta forma ocorrendo à filtragem em todos os sentidos, uma vez que as ondas polarizadas na horizontal pelo primeiro polaróide ficam retidas no segundo polaróide, pois nele passam somente as ondas verticais. Para medir a intensidade de saída da luz quando passada por dois polaróides formando um ângulo (β) é possível utilizar a equação 2.1 (Lei de Malus) onde I_s é a intensidade de saída da luz e I_e é a intensidade da luz de entrada, (UNB, 2000).

$$I_s = I_e \times (\cos \beta)^2 \quad (2.1)^7$$

⁶ Fonte: <http://www.ifi.unicamp.br/~accosta/roteiros/18/nota%2018.html>. Acessado: 25 de março de 2009.

⁷ Fonte: <HTTP://www.unb.br/iq/kleber/EaD/Fisica-4/Aulas/Aula-17/aula-17.html>. Acessado: 24 de março de 2009.

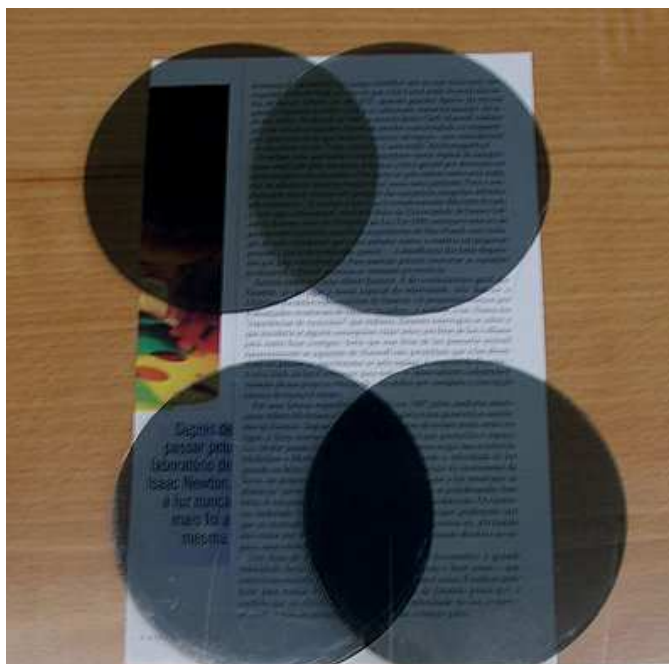


Figura 2-6 - Demonstração dos Polaróides efetuando a polarização da luz⁸

Na figura 2-6, tem-se a demonstração prática do polaróide efetuando a filtragem da luz. Na figura 2-6 aparecem dois pares de polaróides. No primeiro par, mais ao alto, os polaróides estão polarizando no mesmo sentido, permitindo a passagem de luz e possibilitando a visualização do que está escrito do outro lado, no segundo par os polaróides estão efetuando a polarização em sentidos perpendiculares, não sendo possível visualizar o que está escrito do outro lado, pois a luz foi barrada pelos polaróides, (PUCRS, 2009).

2.2 Motor de Passo

2.2.1 Motores Elétricos

Motores elétricos são mecanismos que possuem a característica de transformar energia elétrica em energia mecânica. Este processo é realizado através de campos magnéticos, um produzido por um ímã sólido e outro induzido pela passagem de corrente elétrica, (UNICAMP, 2004).

⁸ Fonte: <http://fisica.fe.up.pt/luz/polarizacao.html>. Acessado: 15 de março de 2009.

Basicamente o funcionamento destes motores é possível graças à combinação destes dois campos magnéticos existentes dentro do dispositivo. O campo magnético fixo é disponibilizado geralmente por imãs, enquanto o campo magnético produzido é gerado pela passagem de corrente elétrica por bobinas contidas no dispositivo, (UNICAMP, 2004).

Sabendo a composição destes motores é possível ter uma idéia simplificada sobre seu funcionamento. Basta imaginar dois imãs, onde um deles gere um campo magnético fixo (imãs) e outro gere um campo magnético que se alterna entre positivo e negativo conforme a corrente que passa por ele (bobinas), fazendo assim com que os campos magnéticos gerados dentro do motor hora se atraia e hora se repulse, fazendo assim com que o eixo gire, (UNICAMP, 2004).

2.2.2 Motores de Passo

Os motores de passo são motores elétricos que trabalham com corrente contínua, e como a maioria dos motores elétricos seu princípio de funcionamento é através do eletromagnetismo. Este tipo de motor foi desenvolvido para ser utilizado em sistemas onde são necessários movimentos precisos para a realização da tarefa, (UFF, 2008).

A diferença entre o motor de passo e outros motores elétricos é muito grande. Enquanto os motores AC e DC comuns trabalham de forma contínua, os motores de passo trabalham com pulsos elétricos mandados de forma sequencial como sinais digitais, (UFF, 2008).

Os motores de passo possuem mais bobinas do que um motor comum, e cada uma é energizada em um momento diferente através dos sinais digitais, fazendo com que cada bobina atraia o eixo do motor quando energizada. Quando estes sinais são enviados de forma sequencial passando de bobina por bobina, ocorre o movimento. Através do envio destes sinais digitais pode-se controlar o movimento do motor tanto na posição, quanto na velocidade. Desta forma, se for necessário que o motor fique parado num determinado ponto, basta manter a bobina referente àquela

localidade energizada e para fazer o motor girar basta energizar uma bobina por vez de forma a seguir a sequência das bobinas do motor, (UFF, 2008).

2.2.3 Passos

O que determina a precisão do motor de passo é justamente o número de bobinas encontradas dentro dele, pois quanto maior o número de bobinas menor será o movimento (passo) gerado a cada comando, o que faz o motor efetuar um número maior de passos para completar uma volta, (UFF, 2008).

A figura 2-7 mostra uma representação da diferença angular entre os passos de um motor de passo com passo de 9° .

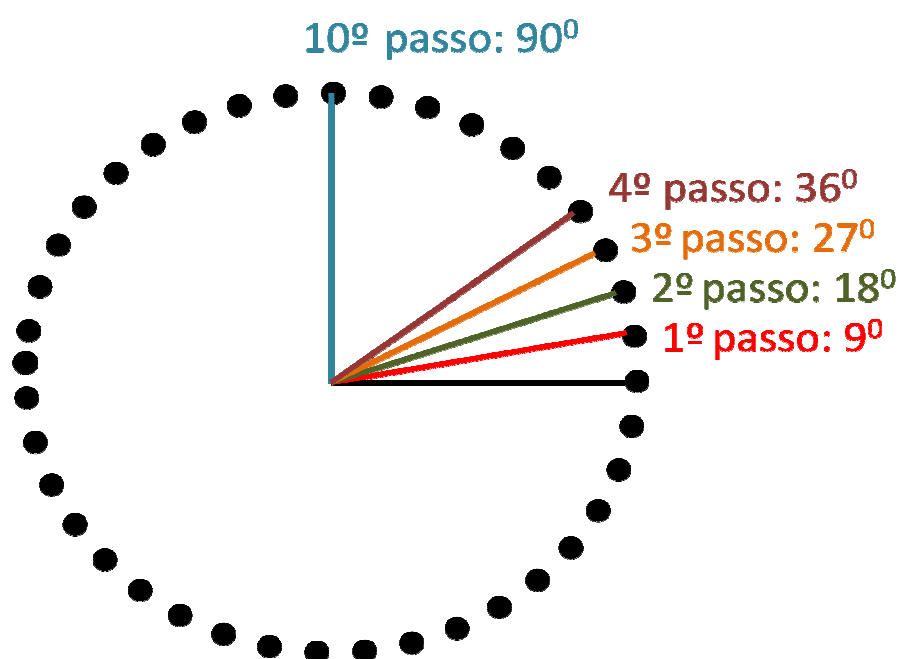


Figura 2-7 – Representação da diferença angular entre os passos do motor de passo⁹

Os movimentos deste motor são gerados através da energização das bobinas contidas em seu interior. Cada vez que uma bobina é energizada ela atrai o rotor

⁹ Fonte: Braga (2009)

fazendo os pólos se alinharem. Efetuando a energização sequencial das bobinas o motor se move conforme a frequência dos sinais emitidos, (UFF, 2008).

Estes passos podem ser gerados de duas maneiras, denominadas passos completos ou meio-passo: os passos completos ocorrem quando apenas uma bobina é energizada a cada movimento alinhando os passo com as bobinas. O meio passo ocorre quando duas bobinas sequenciais são energizadas ao mesmo tempo, gerando um movimento intermediário entre as bobinas, (UFF, 2008).

2.2.4 Tipos de Motores de Passo

Segundo UFF (2008), existe três tipos de motores de passo, os quais possuem como característica a precisão do movimento. São eles:

- Motores de relutância variável
- Motores de imã permanente
- Motores híbridos

2.2.4.1 Motor de Passo por Relutância Variável

A relutância é o fenômeno que um material sofre ao ser exposto a um fluxo magnético. Para girar, este motor se aproveita deste fenômeno uma vez que em seu interior há um material ferroso que sempre busca se alinhar com o ponto onde a relutância é mínima, (UFF, 2008).

Este tipo de motor não possui imã solido o que implica na ausência de torque residual. Ele é composto por um rotor ferroso com múltiplos dentes e um estator com enrolamentos. Sua rotação é dada quando os dentes do estator se atraem pelos pólos magnetizados das bobinas. Por não possuir o imã permanente seu torque é bastante restrito, o que torna sua utilização não muito interessante para algumas aplicações, (UFF, 2008).

Figura 2-8 ilustra o esquema de um motor de passo de relutância variável.

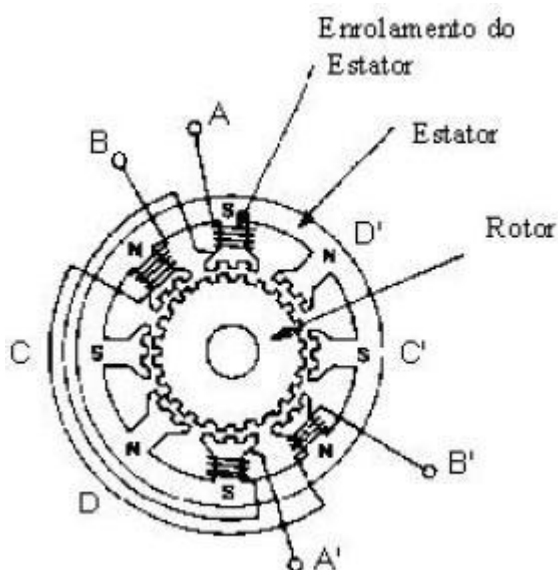


Figura 2-8 – Esquema de um motor de passo por relutância variável¹⁰

2.2.4.2 Motor de Passo de Imã Permanente

Este modelo de motor possui uma boa aceitação nas indústrias para realização das aplicações pelo fato de ser um dispositivo de baixo custo, porém também possui um baixo torque e uma baixa velocidade. Por possuir estas características ele acaba sendo utilizado em vários equipamentos de informática, (UFF, 2008).

Este motor possui uma precisão não tão grande, já que possui ângulos de passo relativamente grandes. Devido à simplicidade de fabricação em grande escala seu custo é relativamente baixo, (UFF, 2008).

Os motores de imã permanente possuem uma variação em seu projeto, que dão origem ao motor de vão axial ou disco, que apresenta um desempenho melhor em medidas maiores já que possui uma inércia baixa. Uma restrição a este motor é quando seu desempenho é exigido ao máximo, pois acaba causando uma baixa inércia, (UFF, 2008).

¹⁰ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

Figura 2-9 ilustra o esquema de um motor de passo de imã permanente.

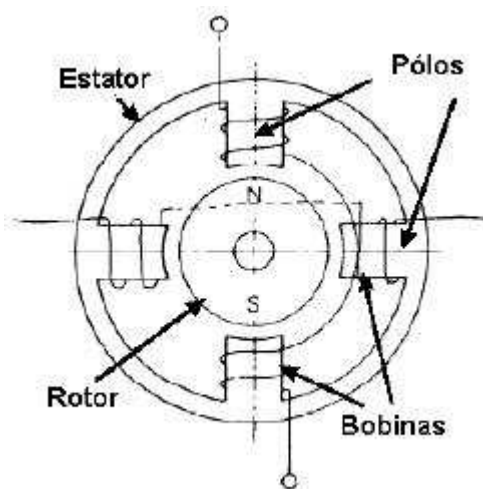


Figura 2-9 – Esquema de um motor de passo de ímã permanente.¹¹

2.2.4.3 Motores de Passo Híbridos

Os motores de passo híbridos são os mais utilizados nas aplicações industriais com um número expressivamente maior que os outros. O nome “híbrido” foi originado pelo fato dele combinar as melhores características dos outros dois motores, utilizando o disco dentado juntamente com ímãs sólidos e também os estatores magnetizados pela passagem de corrente DC, (UFF, 2008).

Este tipo de motor possui uma precisão que está entre $3,6^\circ$ e $0,9^\circ$ entre os passos o que requer de 100 a 400 passos a cada volta (360°), (UFF, 2008).

Figura 2-10 ilustra o esquema de um motor híbrido.

¹¹ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

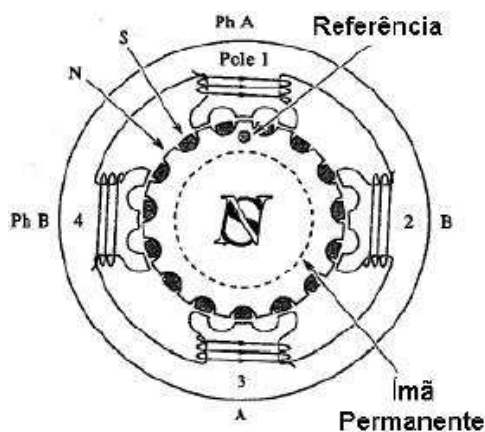


Figura 2-10 – Esquema de um motor de passo híbrido¹²

2.2.5 Formas de Operação

Conforme UFF (2008), as formas de operação de um motor de passo podem ser:

- Unipolar;
- Bipolar.

2.2.5.1 Motores Unipolares

Nos motores unipolares, geralmente as bobinas existentes são ligadas diretamente a um aterramento comum, sendo necessário para movimentá-lo apenas enviar os sinais digitais diretamente as bobinas. Desta forma sua configuração se torna mais simples de trabalhar já cada bobina recebe o sinal sem ter que se preocupar com sentido da corrente, (UFF, 2008).

Figura 2-11 demonstra bobinas ligadas ao terra comum.

¹² Fonte: www.ebah.com.br/motor-de-passo-pdf-pdf-a5479.html. Acessado em 20 de março de 2009

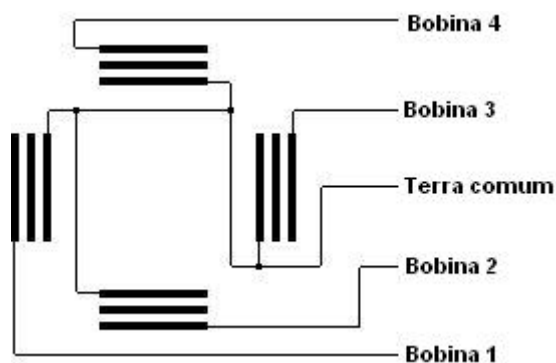


Figura 2-11 – Demonstração das ligações das bobinas de um motor de passo unipolar¹³

Durante a configuração do motor de passo unipolar pode-se escolher tanto o passo inteiro como o meio passo, (UFF, 2008).

Para girar um motor de passo é necessário energizar as bobinas de forma sequencial. Quando ocorre este envio de sinal elétrico sequencial nos fios do motor de passo o ímã sólido contido no seu eixo faz com que o campo magnético dele se alinhe ao campo magnético gerado pela passagem da corrente elétrica pela bobina, assim gerando a atração necessária para girar o motor. A figura 2-12 ilustra o esquema de um motor de passo unipolar com movimento por passo inteiro, onde observa-se a atração gerada pelo campo magnético criado pela passagem de corrente pela bobina e o pólo magnético de sentido inverso existente no ímã sólido do eixo do motor de passo, (UFF, 2008).

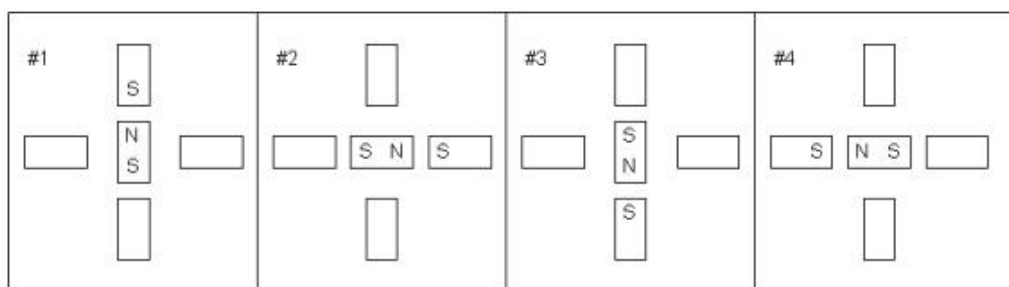


Figura 2-12 – Esquema de um passo inteiro do motor de passo unipolar¹⁴

¹³ Fonte: Braga (2009)

¹⁴ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

Os movimentos de meio passo, bem como os de passo inteiro também são gerados pelos sinais elétricos sequenciais, a diferença entre eles é caracterizada por um movimento intermediário provocado pelo envio de um sinal elétrico a duas bobinas sequenciais (a que está energizada e a que seria energizada para dar continuidade ao movimento do motor) ao mesmo tempo durante a passagem de um passo para outro. Enviando o sinal elétrico as duas bobinas é criado um movimento intermediário que dá origem ao meio passo. A figura 2-13 ilustra esquema de um motor de passo com movimento por meio passo, (UFF, 2008).

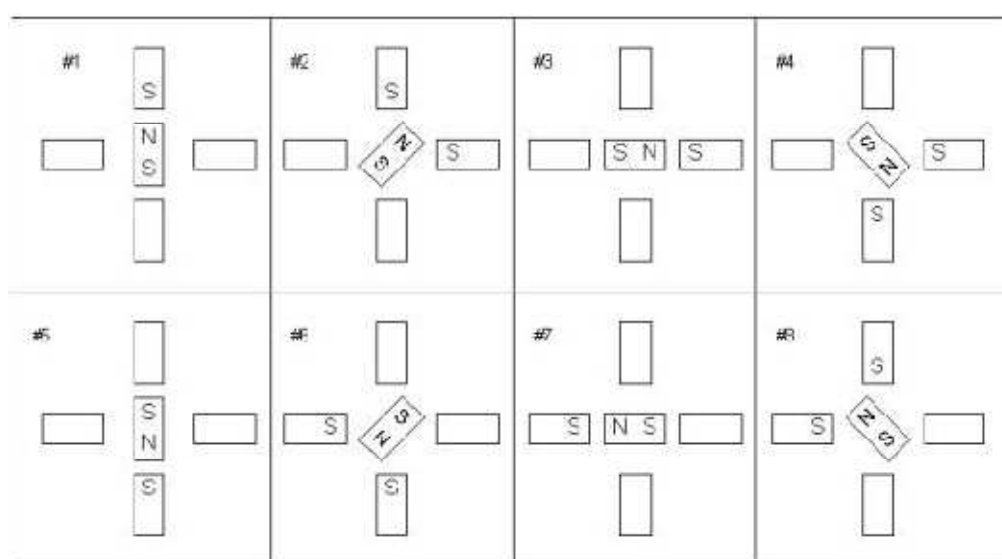


Figura 2-13 – Esquema de meio passo do motor de passo unipolar¹⁵

2.2.5.2 Motores Bipolares

Nos motores de passo bipolares as bobinas não são totalmente independentes uma das outras e também não possuem o aterramento comum como nos unipolares, o que torna seu controle um pouco mais complexo, uma vez que é necessário inverter o sentido dos sinais elétricos que atravessam as bobinas para completar um ciclo de movimentos do motor de passo. Este processo de inversão no sentido dos sinais elétricos é necessário pelo fato de suas bobinas serem ligadas em pares, sendo preciso efetuar a inversão para modificar o pólo do campo magnético gerado

¹⁵ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

pela passagem da corrente elétrica nas bobinas. Quando é feita esta inversão de sentido do sinal elétrico estamos automaticamente mudando o pólo do campo magnético gerado pela bobina. Figura 2-14 ilustra ligações entre pares de bobinas, (UFF, 2008).

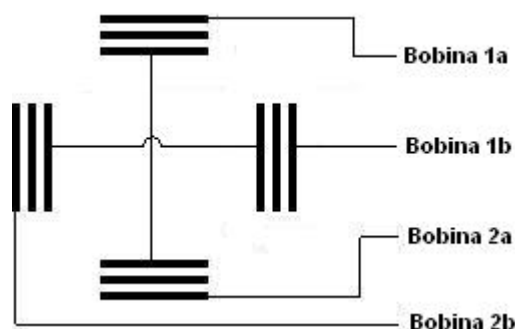


Figura 2-14 – Demonstração das ligações das bobinas de um motor de passo bipolar¹⁶

Semelhante aos motores de passo unipolares, os bipolares também possuem movimentação por passo inteiro e meio passo. A sequência nos sinais elétricos que devem ser enviados ao motor de passo são iguais nos unipolares e bipolares. O que diferencia um tipo do outro são os campos magnéticos gerados pela passagem da corrente elétrica pelas bobinas, enquanto o unipolar gera um campo magnético de por vez o bipolar gera dois campos magnéticos, um em cada bobina, sendo que possuem sentido de pólos magnéticos diferentes. Por existir dois campos magnéticos gerados pelas bobinas pela passagem de corrente elétrica o bipolar apresenta um torque maior que o unipolar, pois assim os dois pólos magnéticos gerados serão atraídos pelos pólos opostos do ímã sólido existentes no eixo do motor de passo, as figuras 2-15 e 2-16 ilustram respectivamente o esquema da movimentação do motor de passo bipolar por passo inteiro e meio passo, (UFF, 2008).

¹⁶ Fonte: Braga (2009)

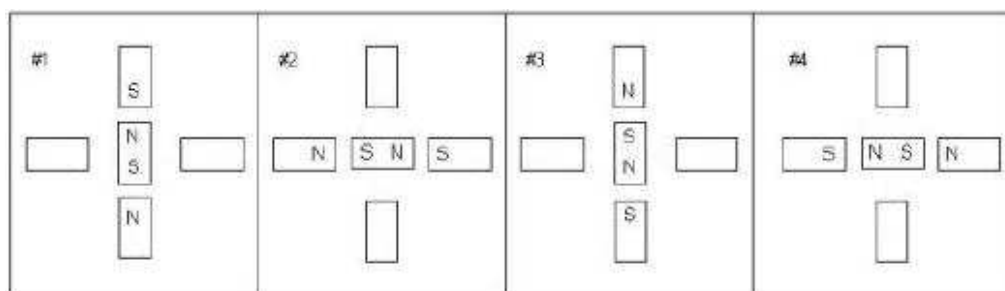


Figura 2-15 – Esquema de passo inteiro do motor de passo bipolar¹⁷

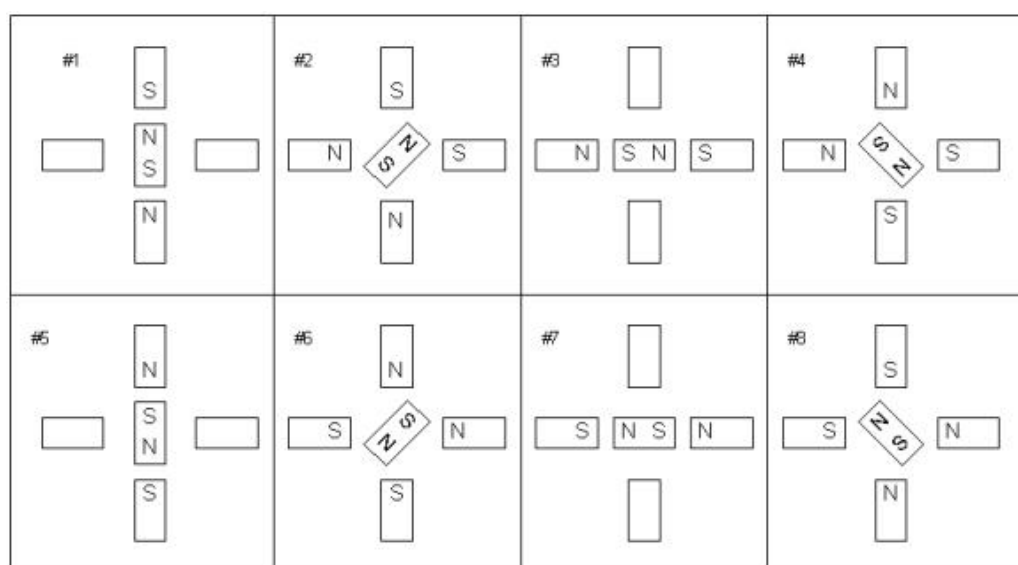


Figura 2-16 – Esquema de meio passo do motor de passo bipolar¹⁸

Comparando as figuras 2-12 e 2-15 com as figuras 2-13 e 2-16, é possível perceber a diferença de funcionamento entre motores de passo unipolares e bipolares. Os unipolares por possuírem as bobinas totalmente independentes uma das outras, geram apenas um campo magnético, enquanto no bipolar são gerados dois campos magnéticos de polaridades opostas desta forma trazendo a ele a vantagem de possui um torque maior que o unipolar, (UFF, 2008).

¹⁷ Fonte: Braga (2009)

¹⁸ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

2.2.6 Controle do Motor de Passo

Os motores de passo trabalham através de sinais digitais “0” e “1”, onde “0” é a ausência de energia e “1” é o envio de sinal elétrico. Pensando em sinais digitais pode-se estabelecer uma lógica de funcionamento para a sequência de energização das bobinas, conseqüentemente gerando os passos e colocando o motor de passo em movimento, (UFF, 2008).

Na tabela 2-1 é mostrada a forma de configuração dos sinais gerados para passo completo. Esta primeira forma possui um consumo menor de energia, porem possui um torque e precisão menores, (UFF, 2008).

Tabela 2-1 – Tabela passo inteiro 1¹⁹

Nº do Passo	B0	B1	B2	B3
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

A tabela 2-2 informa como deve ser a configuração dos sinais enviados para motores de passo de forma a movimentá-los com passo completo através de duas bobinas. Trabalhar com duas bobinas sendo energizadas ao mesmo tempo, permite a vantagem de obter um torque maior, porém o consumo de energia se torna maior, (UFF, 2008).

Tabela 2-2 – Tabela passo inteiro 2²⁰

Nº do Passo	B0	B1	B2	B3
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

¹⁹ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

²⁰ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

Outra forma de emissão de sinais é a mostrada na tabela 2-3, que informa como é a lógica para a configuração de um motor de passo sendo movido por meio passo. Esta forma de movimentação possui uma precisão maior que as duas apresentadas anteriormente e um torque pouco maior que o segundo caso. Isto ocorre pelo fato de possuir o meio passo, que faz com que o passo produzido pelas bobinas seja menor, utilizando melhor os campos magnéticos, (UFF, 2008).

Tabela 2-3 – Tabela meio passo²¹

Nº do Passo	B0	B1	B2	B3
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Para movimentar o motor de passo utilizado neste projeto foi utilizado a sequencia de sinais informados na tabela 2-1, assim classificando o motor de passo no projeto como: híbrido, bipolar e movimentado por passo inteiro.

2.3 Microcontrolador 8051 AT89S52

A Intel lançou o primeiro microcontrolador no ano de 1976, esse primeiro dispositivo recebeu o nome de 8048, e posteriormente passou a definir uma família de microcontroladores, a família MCS-48. O microcontrolador 8051 foi lançado quatro anos depois, em 1980, sendo uma evolução do 8048, e passou a definir a família MCS-51. O 8051 teve grande aceitação e mesmo tendo passados mais de 20 anos, ele ainda está presente no mercado. Em 1982 a Intel lançou o 8096, um

²¹ Fonte: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado em 20 de março de 2009

microcontrolador de 16 bits, formando assim a família MCS-96, (ZELENOVSKY, 2005).

O primeiro lançamento da Intel da família 8051 foi o 8051 com ROM interna programável de fábrica, logo em seguida lançou o 8751 com EPROM interna programável pelo próprio usuário e posteriormente o 8031 com chips de EPROM externa. Posteriormente novos chips foram adicionados a esta família, o 8052, que com um timer a mais que o 8051 e com as seguintes variações: com EPROM (8752); sem ROM/EPROM (8032) e uma versão chamada de 8052 – que já vem interpretador de programas em linguagem Basic. Os microcontroladores desta família podem ser fabricados por diversas empresas, dentre elas: a Siemens, Advanced Microcontroller Device, Fujitsu, Philips e Oki, (NICOLOSI, 2000).

Este projeto utiliza um microcontrolador AT89S52 da família 8051. A decisão pelo uso deste componente foi influenciada pelas facilidades de programação (aceita programação em linguagem de alto nível), facilidade de ser encontrado no mercado, baixo custo, compatibilidade com todas as variações da família 8051, boa quantidade de fabricantes e boa velocidade na execução das tarefas, (MARINHO, 2001).

A figura 2-17 ilustra o diagrama de bloco do microcontrolador 8051 representando os componentes básicos contidos internamente, descritos abaixo segundo Nicolosi (2008):

- RAM interna de uso geral de 128 bytes e 128 bytes correspondentes aos registradores especiais;
- ROM interna de 4Kbytes;
- 4 Ports de I/O;
- 2 Timers de 16 bits;
- 1 Interface serial;
- Capacidade de 64 Kbytes de endereçamento externo de ROM;
- Capacidade de 64 Kbytes de endereçamento externo de RAM;
- Processador “Booleano” (opera com bits);

- Ciclos típicos de instrução de 1 e 2 μ s a clock de 12 MHz;
- Instrução direta de divisão e multiplicação
- Entradas de interrupção externa.

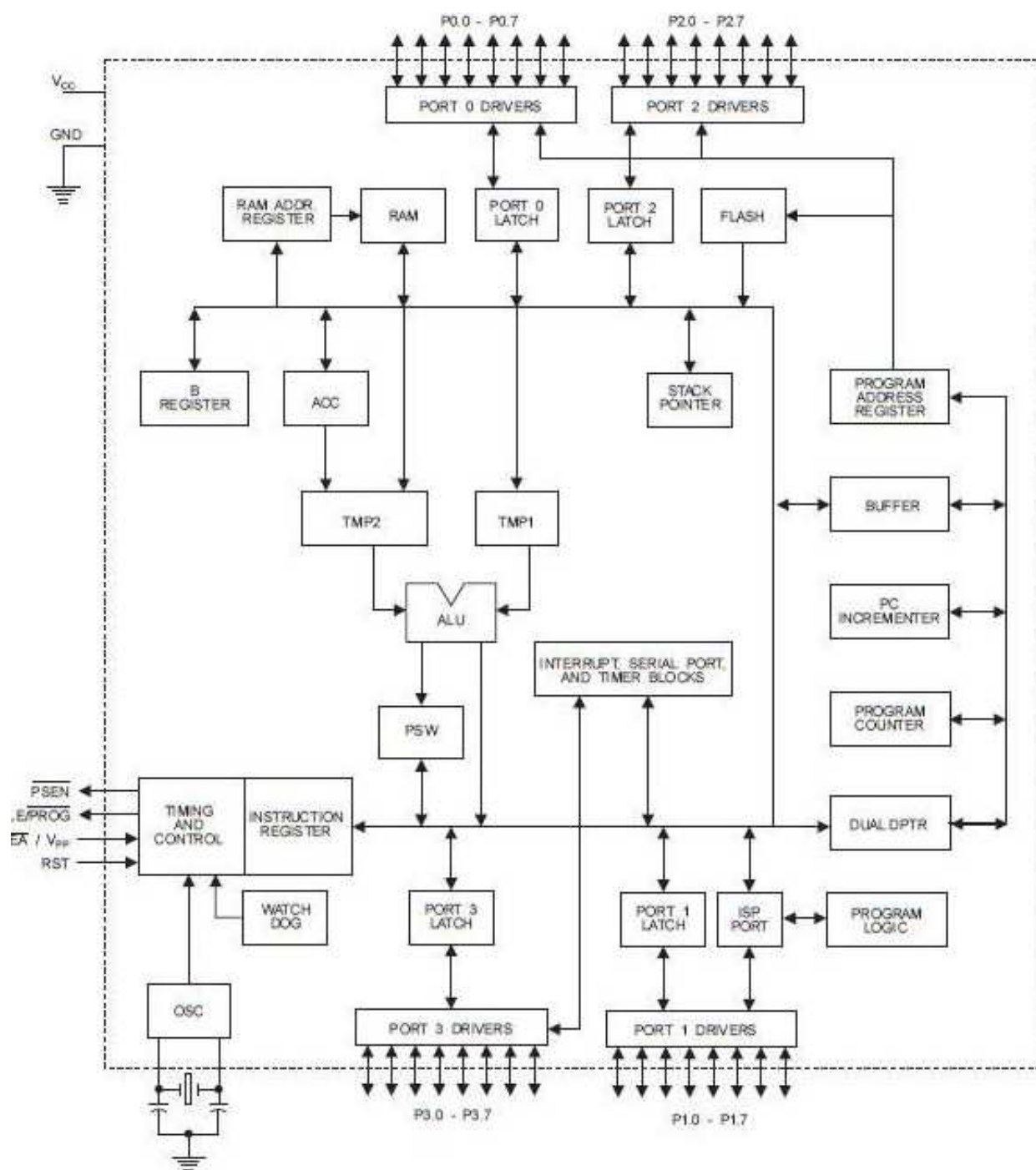


Figura 2-17 – Diagrama do microcontrolador 8051²²

²² Fonte : Datasheet do microcontrolador AT89S52.

2.3.1 Pinagem do Microcontrolador 8051 AT89S52

A figura 2-18 ilustra os pinos do microcontrolador AT89S52.



Figura 2-18 – Pinagem microcontrolador 8051²³

Segundo Marinho (2001), apesar do número de pinos no microcontrolador 8051 ser alto, é simples entendê-los, pois suas funções são bem definidas. Seguem abaixo:

- Pinos de 1 a 8 – Referem-se à porta P1, esta porta é utilizada tanto para emissão de sinais como para o recebimento. Pode-se adicionar até 4 portas TTL-LS em cada pino;
- Pino 9 – Pino responsável pelo reset do sistema. Este processo ocorre quando é enviado para o pino um nível lógico alto por pelo menos 2 ciclos;
- Pinos 10 ao 17 – Referem-se a Porta P3 que possui as mesmas características da Porta P1 mas com algumas funções extras, como:

Pino 10 (RXD/DATA): Entrada de dados emitidos por conexão serial;

Pino 11 (TXD/CLOCK): Saída para interpretação de conexão serial;

²³ Fonte : Datasheet do microcontrolador AT89S52.

Pino 12 (INT0): Interrupção externa 0, ativa em nível lógico baixo;

Pino 13 (INT1): Interrupção externa 1, acionado também por nível lógico baixo;

Pino 14 (T/C0): Temporizador/contador de eventos;

Pino 15 (T/C1): Entrada de um temporizador/contador externo;

Pino 16 (WR): Sinalizador de escrita de dados externa;

Pino 17 (RD): Sinalizador de entrada de dados externos.

- Pinos 18 e 19 – São XTAL1 e XTAL2 respectivamente;
- Pino 20: VSS (aterramento do Microcontrolador 8051);
- Pinos 21 ao 28 – Porta P2.0 a P2.7 possui as mesmas funcionalidades da porta P1 mas sendo utilizados como endereçamentos mais significativos para que completem juntamente com a parte menos significativa do barramento o endereçamento de 16Bits;
- Pino 29 – Sinalizador para leitura de programa gravado em memória de programa externa;
- Pino 30 – Saída para habilitação do *latch* de endereços, que separa o barramento menos significativo do mais significativo assim disponibilizando o endereçamento de 16 bits através da junção das portas P0 e P2;
- Pino 31 – Chave seletora de memória. Se inserido um nível lógico baixo a CPU executa somente as instruções contidas em uma Memória de Programa externa. Se inserido um nível lógico alto a CPU executa somente as instruções contidas em uma Memória de Programa interna.
- Pino 32 ao 39 – Porta P0 é o endereçamento menos significativo contendo 8 bits bidirecionais e isentos de resistores internos.
- Pino 40 – VCC.

O microcontrolador AT89S52 da família 8051 será utilizado no projeto para executar os comandos gerados na programação, assim controlando todo o sistema.

O microcontrolador AT89S52 será acoplado ao circuito conforme mostra a figura 4-11.

2.4 L298N

O CI (Circuito Integrado) L298N, mostrado na figura 2-19, possui em sua configuração duas “Ponte H”, o que possibilita cada componente controlar motores de passo bipolar que possuem quatro fios e que utilizam tensões de até 46V e corrente de até 2A através de sinais de entrada TTL (*Transistor-Transistor Logic*). Neste projeto, o CI terá como finalidade receber o sinal do microcontrolador e liberar a corrente elétrica para alimentar o motor de passo bipolar, (DATASHEET CI L298N).



Figura 2-19 – CI L298N²⁴

Este CI dispõe de quinze pinos para funcionamento e configuração, conforme mostra a figura 2-20. Na ilustração os pinos maiores são os pinos retos e os menores são os pinos dobrados de fábrica, (DATASHEET CI L298N).

²⁴ Fonte: http://www.akcesoria.cnc.info.pl/elektronika_cnc.htm. Acessado em 17 de abril de 2009

2.4.1 Ponte H

“Ponte H é um circuito eletrônico que permite que um motor gire tanto para um sentido quanto para o outro. Estes circuitos são geralmente utilizados em robótica estão disponíveis em circuitos prontos ou podem ser construídos por componentes. O nome ponte H é dado pela forma que assume o circuito quando montado”, (UFF, 2008).

Como visto, o CI L298N possui em sua composição duas Pontes H o que permite controlar um motor de passo bipolar de até quatro fios com um único componente. Conforme colocado no item 2.2 deste capítulo, nos motores de passo bipolares é necessário efetuar a mudança de pólo das bobinas para que o movimento se complete, (UFF, 2008).

A Ponte H entra justamente para permitir esta mudança de pólo das bobinas do motor bipolar, pois este circuito possui uma alimentação e um aterramento continuamente presente, aguardando apenas a emissão de um sinal para liberar a passagem da corrente pelo motor no sentido correto, (UFF, 2008).

Pode-se observar na figura 2-22 que ao enviar o sinal elétrico na entrada “IN 1” é liberada a passagem de corrente elétrica nos pontos A e D fazendo com que ela atravesse o motor no sentido indicado pelas setas amarelas e quando o sinal elétrico é enviado para a entrada “IN 2” a corrente elétrica é liberada para passar pelos pontos B e C fazendo com que a corrente elétrica atravesse o motor no sentido indicado pelas setas verdes. Desta maneira é feito o controle eficaz na inversão do sentido da corrente elétrica, que é necessário no motor bipolar, permitindo que o motor de passo execute o ciclo completo de rotação.

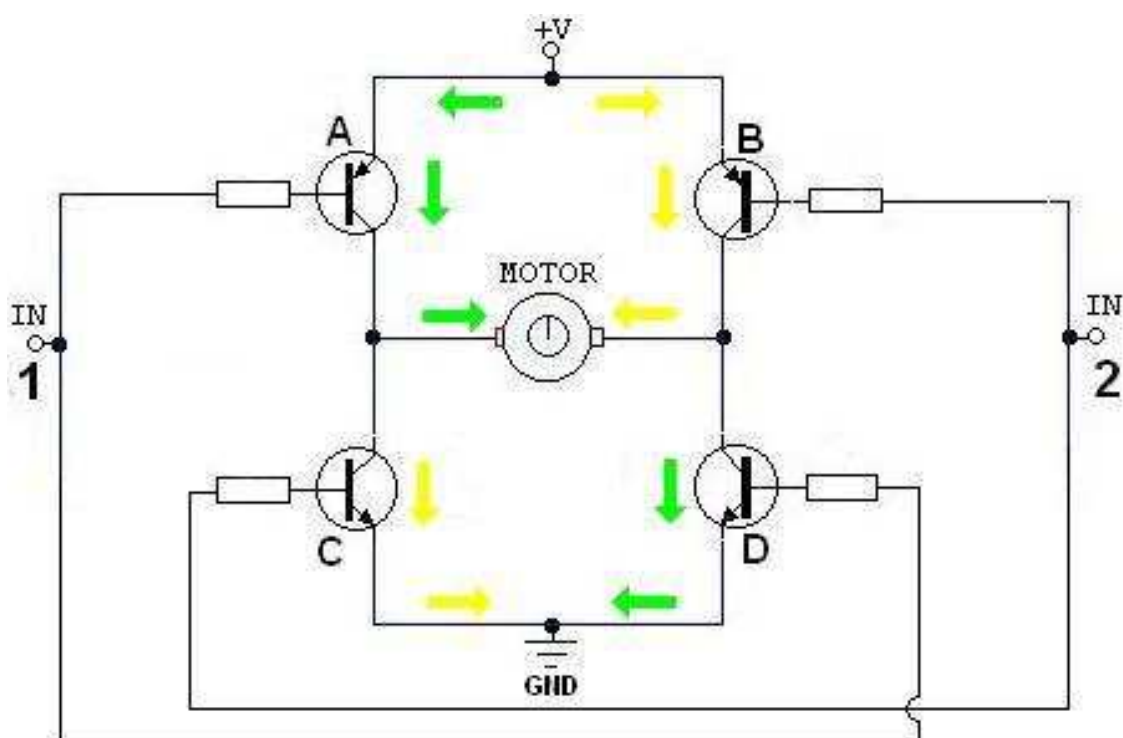


Figura 2-22 – Esquema Ponte H²⁷

2.5 Ferramentas Utilizadas

Este projeto utiliza alguns *softwares* para o desenvolvimento do *hardware*. Para a simulação do circuito foi utilizado o *software Proteus 7* que se mostrou uma excelente ferramenta diante de todas as funcionalidades e praticidades apresentadas. Este *software* possui um banco de dados bastante extenso com diversos componentes eletrônicos que permite a montagem de todo o circuito, aceitando o *software* desenvolvido para o microcontrolador 8051 e verificando seu funcionamento no *hardware*. O único inconveniente encontrado é ser um *software Shareware* (pago), (LABCENTER, 2009).

Outra ferramenta utilizada foi o compilador *ConText* que compila o arquivo C para o formato “.HEX” que é um formato reconhecido pelo microcontrolador AT89S52, (SUPERDOWNLOADS, 2009).

²⁷ Fonte: Braga(2009)

Também foi utilizada a ferramenta *Jude/Communit*, que é uma versão gratuita do desenvolvedor das ferramentas Jude, para a criação do Fluxograma do software que compõem o projeto, (JUDE, 2009).

3 IMPLEMENTAÇÃO DO *HARDWARE* E DO *SOFTWARE*

O *hardware* foi desenvolvido para girar a armação que prende a película polarizadora móvel nos dois sentidos, conseguindo assim o efeito do aumento ou redução do nível de transparência ocasionado pelo efeito da polarização.

Este capítulo descreve as ligações aplicadas entre os componentes para que o *hardware* envie o sinal para o motor de passo para que gire a armação móvel da janela.

3.1 Descrição do *Hardware*

Para a construção do *hardware* é necessário os seguintes componentes: um microcontrolador 8051 AT89S52, dois capacitores de 33pF e um de 10μF, quatro resistores de 10kΩ e um de 470Ω, cristal de frequência de 27Mhz, fonte de alimentação de 5.5V, um CI L298N, um motor de passo híbrido e bipolar, um *display* LCD 16X2, uma placa para montagem do circuito e *push buttons*.

Para a montagem do *hardware* foi seguida a seguinte ordem: começando pelo motor de passo, passando pelo CI L298N e finalmente chegando ao microcontrolador onde também é ligado o *display*, dois *push buttons* para controle do dispositivo e dois emissores de sinal que confirmam o movimento da janela.

3.1.1 Motor de Passo Utilizado

Este projeto utiliza um motor de passo híbrido e bipolar (marca Tamagawa modelo TS 3218 N814) para ter um torque maior comparado com os outros dois

tipos de motores de passo (relutância variável e imã permanente), conforme especificado pela UFF (2008).

A tabela 3-1 mostra os passos deste motor da forma em que foram conectados ao projeto, especificando a ordem dos passos recebidos pelo motor.

Tabela 3-1 – Esquema de passos utilizados no projeto²⁸

Nº do Passo	Azul	Branco	Vermelho	Amarelo
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Este motor trabalha com tensões máximas de 6.9 Volts e sua corrente nesta tensão é de 0.32A, mas no projeto é utilizado apenas 5.5 Volts, fazendo sua corrente cair para 0.25A, para igualá-la a tensão do hardware montado, facilitando a montagem do circuito e dispensando componentes extras necessários para efetuar o ganho de tensão para atingir o nível máximo aceito pelo motor de passo.

Este motor possui um alto nível de precisão, 0.9° por passo, ou seja, para o motor efetuar uma volta completa (360°) seriam necessários 400 passos.

A figura 3-1 mostra a parte de trás do motor com ilustração de algumas especificações.

²⁸ Fonte: Braga (2009)



Figura 3-1 – Parte de trás do motor de passo utilizado.²⁹

As figuras 3-2 e 3-3 mostram como é o motor de passo fechado e aberto respectivamente.

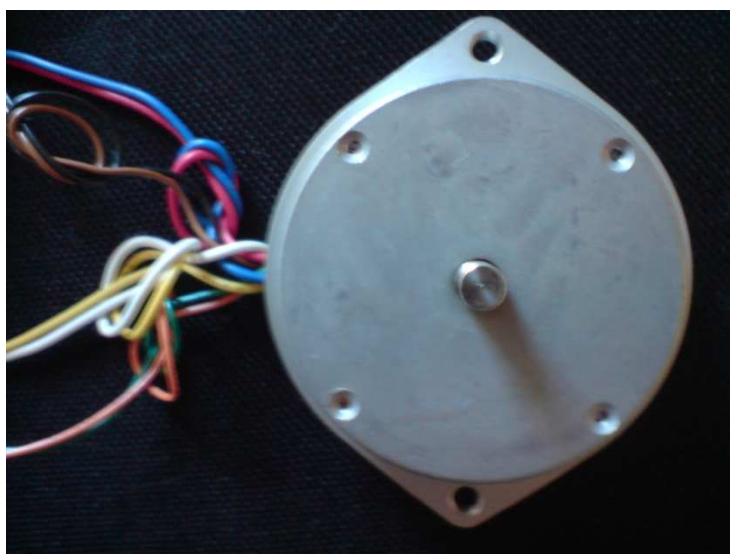


Figura 3-2 – Motor de passo fechado³⁰

²⁹ Fonte: Braga (2009)

³⁰ Fonte: Braga (2009)

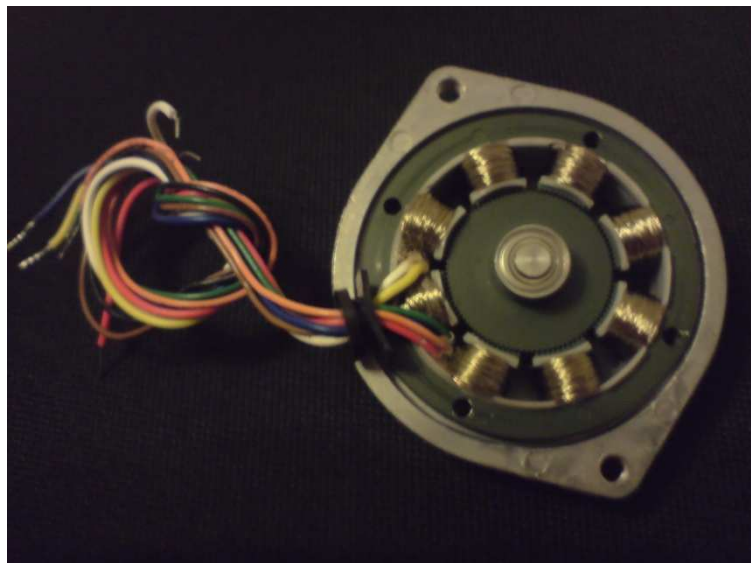


Figura 3-3 – Motor de passo aberto³¹

3.1.2 CI L298N

Conforme visto no capítulo 2, o CI L298N é responsável por permitir que o motor de passo dê continuação a sua rotação por dispor de duas Pontes H em sua composição.

Abaixo segue descrição das ligações dos pinos ao circuito para funcionamento do CI L298N por ordem numérica conforme ilustrado na figura 2-20 disponível no capítulo 2:

- Pino 1: Ligado ao terra do circuito;
- Pino 2: Ligado aos primeiros dois fios referentes ao primeiro passo do motor de passo (Azul);
- Pino 3: Ligado ao segundo par de fios do motor de passo, referentes ao segundo passo do motor de passo (branco);

³¹ Fonte: Braga (2009)

- Pino 4: Neste pino liga-se a tensão que alimenta o motor de passo para que efetue o movimento. No caso deste projeto 5.5Volts;
- Pino 5: Pino referente ao sinal enviado pelo microcontrolador para que o motor efetue o movimento referente a o primeiro passo;
- Pino 6: Este pino é ligado na tensão de 5.5 Volts para a habilitação da Ponte H “A”;
- Pino 7: Pino referente ao sinal enviado pelo microcontrolador para que o motor efetue o movimento referente a o segundo passo;
- Pino 8: Terra do CI L298N;
- Pino 9: Pino que habilita da recepção de sinais TLL, para que o CI possa interpretar os sinais emitidos pelo microcontrolador;
- Pino 10: Pino referente ao sinal enviado pelo microcontrolador para que o motor efetue o movimento referente a o terceiro passo;
- Pino 11: Este pino é ligado também na tensão de 5.5 Volts desta vez para a habilitação da Ponte H “B”;
- Pino 12: Pino referente ao sinal enviado pelo microcontrolador para que o motor efetue o movimento referente a o quarto passo;
- Pino 13: Pino que recebe o sinal enviado pelo microcontrolador para que a corrente seja liberada para alimentar motor de passo (fio vermelho) para que o terceiro passo seja executado;
- Pino 14: Este pino envia ao motor de passo o sinal recebido do microcontrolador para que o quarto passo seja executado (amarelo);
- Pino 15: Ligado ao terra do circuito.

Com as ligações apresentadas acima tem-se o circuito eletrônico que faz a intermediação entre o motor de passo e o microcontrolador instalado, funcionando de forma manual através de *push buttons* (instalados nas portas input do CI L298N) que simulam o envio do sinal no microcontrolador cada vez que acionados. Este procedimento torna-se necessário para ter certeza e garantir o funcionamento do motor antes de começar a trabalhar com o microcontrolador.

Abaixo a figura 3-4 mostra como fica o circuito montado para o controle do motor de passo de forma manual (para melhor verificação da movimentação do motor, foi colocado um pequeno pedaço de fio em seu eixo), com a confirmação do funcionamento, é possível a montagem e configuração do microcontrolador no circuito.

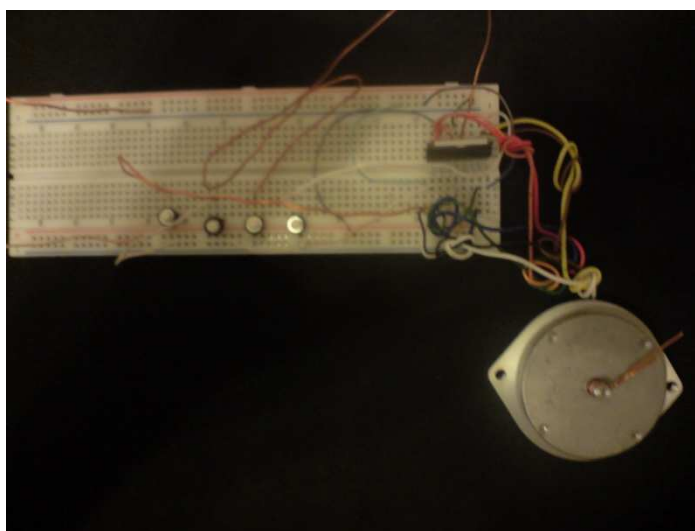


Figura 3-4 – Circuito de ligação entre CI L298N e motor de passo com acionamento por *push buttons*.³²

3.1.3 Ligações do Microcontrolador 8051

O microcontrolador é o componente responsável por receber as informações e controlar o motor de passo para que a janela execute a tarefa proposta.

O microcontrolador tem a função de receber a informação do usuário e retransmiti-la ao motor de passo para execução da tarefa. Ele também envia

³² Fonte: Braga (2009)

informações do sistema para o usuário através de um *display* de LCD de 16X2 (duas linhas, cada uma com dezesseis colunas para escrita).

As ligações elétricas com o microcontrolador são as seguintes:

- Porta P0.0: é ligada ao *push button* disponível ao usuário, responsável por diminuir o nível de transparência da janela;
- Porta P0.1: é conectada ao segundo *push button* disponível ao usuário. Este possui a função de aumentar o nível de transparência da janela;
- Porta P0.2: é responsável por receber o sinal que confirma que a janela alcançou o nível de transparência desejado pelo usuário;
- Porta P0.3: recebe o sinal de confirmação de que a janela alcançou o nível máximo de transparência na inicialização do sistema. A janela terá um sistema contra falhas, pois ao ter seu fornecimento de energia interrompido a janela pode perder seu referencial ficando vulnerável a interferência humana ou qualquer outra, devido à ausência de torque no motor de passo. Esse sistema contra falhas considera que toda vez que existir uma interrupção no fornecimento de energia, a janela retorna ao nível máximo de transparência para que possa voltar a se movimentar com precisão. Este sinal tem a função de informar ao sistema o momento em que a janela chega ao nível máximo de transparência para dar início a interatividade com o usuário.
- Porta P1: é responsável pelo envio de dados que correspondem as letras que aparecem no LCD. O envio de dados se dá através de 8 bits, assim os fios do 7 ao 14 do *display* são ligados respectivamente aos pinos P1.0 ao P1.7 do microcontrolador.
- Porta P2.0: é responsável por enviar o sinal que posiciona o motor de passo no primeiro passo;

- Porta P2.1: é responsável por enviar o sinal que posiciona o motor de passo no segundo passo;
- Porta P2.2: é responsável por enviar o sinal que posiciona o motor de passo no terceiro passo;
- Porta P2.3: é responsável por enviar o sinal que posiciona o motor de passo no quarto passo;
- Porta P3.6: é responsável por enviar o sinal que habilita a escrita no *display*;
- Porta P3.7: é responsável por informar ao *display* se o dado que está sendo emitido é um comando ou uma informação a ser escrita nele;
- Porta EA: é ligada juntamente à alimentação do microcontrolador para que o CI busque as instruções em sua memória interna;
- Portas XTAL 1 e 2: são conectadas por um cristal de frequência para determinar o *clock* de processamento do microcontrolador. Para o microcontrolador 8051 AT89252 (utilizado no projeto) as frequências aceitas estão entre 3 e 33 MHz. Este projeto conta com um cristal de 27MHz criando um delay de 24ms entre um passo e outro, fazendo com que um giro do motor demore 9,6s, o que é suficiente para o protótipo criado;
- Porta Vcc: responsável pela alimentação do microcontrolador;
- Porta GND: aterramento do sistema

A figura 3-5 mostra como deve ser efetuada a ligação dos cristais de frequência conectados a porta XTAL1 E XTAL2.

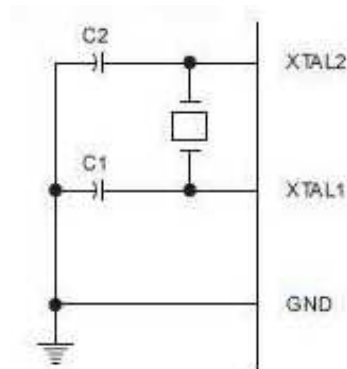


Figura 3-5 – Esquema de ligação do Cristal de Frequência.³³

A figura 3-6 mostra como devem ser efetuadas as ligações, dos *push buttons* e dos emissores de sinais localizados na janela, ao microcontrolador para que não coloque as portas em curto.

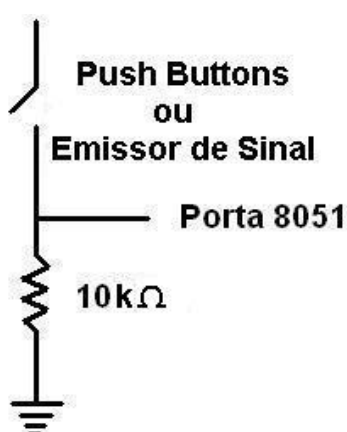


Figura 3-6 – Esquema de ligação dos push buttons e emissores de sinal³⁴

3.1.3.1 Esquema dos Dispositivos Emissores de Sinal

Este projeto conta com dois sistemas emissores de sinais existentes no protótipo. Estes sistemas possuem a função de enviar ao microcontrolador sinais informando ao microcontrolador que o movimento solicitado ao sistema foi concluído com sucesso.

³³ Fonte : Datasheet do microcontrolador AT89S52

³⁴ Fonte: Braga (2009)

Este sistema funciona através do contato entre um ponto metálico que está fixado à armação de sustentação das películas e algum dos pontos metálicos fixados na outra armação que sustenta as roldanas e o motor de passo. Este contato permite a passagem de corrente elétrica a qual se trata do sinal informado ao microcontrolador de que o movimento solicitado foi concluído.

A figura 3-7 mostra o funcionamento deste sistema.

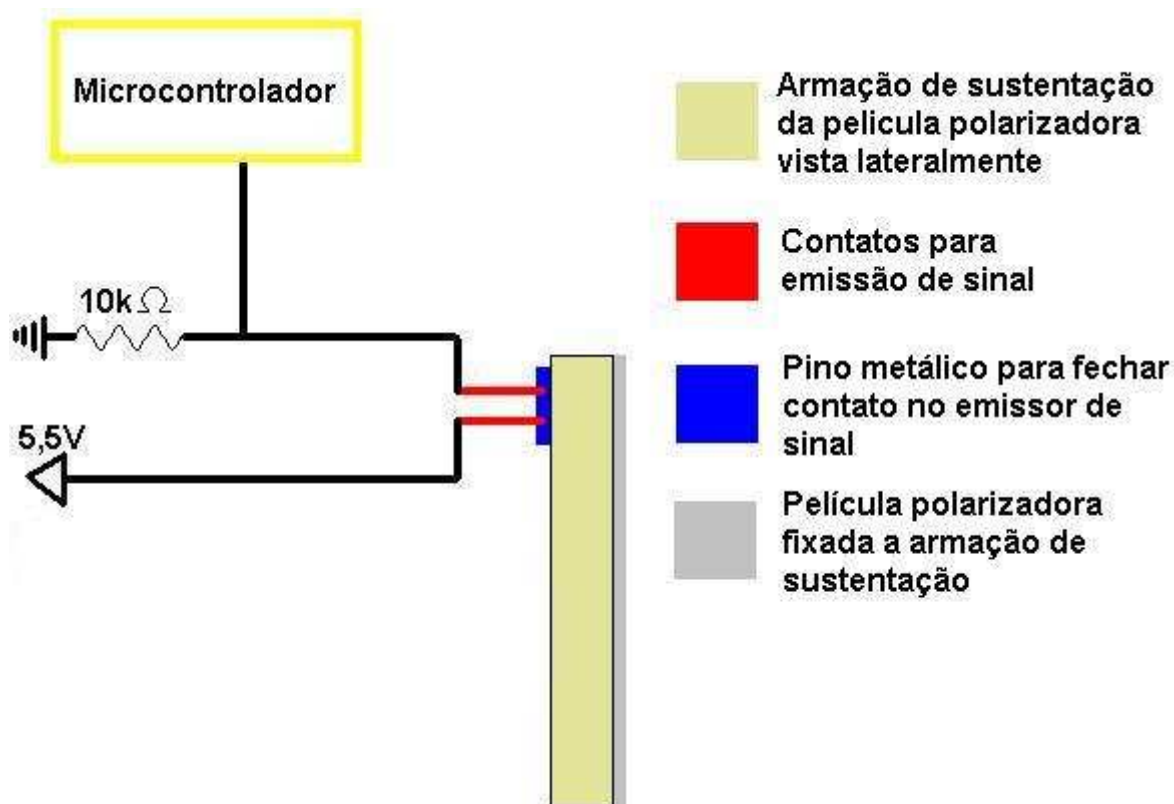


Figura 3-7 – Esquema do sistema de emissão de sinal³⁵

No protótipo existem dois emissores de sinais, um é o “Emissor de Sinal Confirma o Movimento” que informa ao microcontrolador que a mudança do nível de transparência solicitada foi efetuada corretamente e o outro é o “Emissor de Sinal Finaliza Inicialização” que emite o sinal para o microcontrolador informando que a janela se encontra no nível máximo de transparência. Este emissor de sinal é utilizado somente na inicialização do sistema, caso ocorra imprevistos como a falta de energia, (está função é explicada no item 3.1.3 desta monografia).

³⁵ Fonte: Braga (2009)

Estes emissores de sinal são visualizados na figura 4-12 que mostra a parte de trás das armações de sustentação das películas polarizadoras, roldanas e motor de passo, nesta figura tem-se dois pares de fios conectados, onde o par de fio fixado na parte superior faz parte do “Emissor de Sinal Finaliza Inicialização” e o par fixado a direita são pertencentes ao sistema “Emissor de Sinal Confirma Movimento”

Na figura 3-8 tem-se a representação da armação de sustentação da película móvel com os pinos de fechamento de curto fixados.

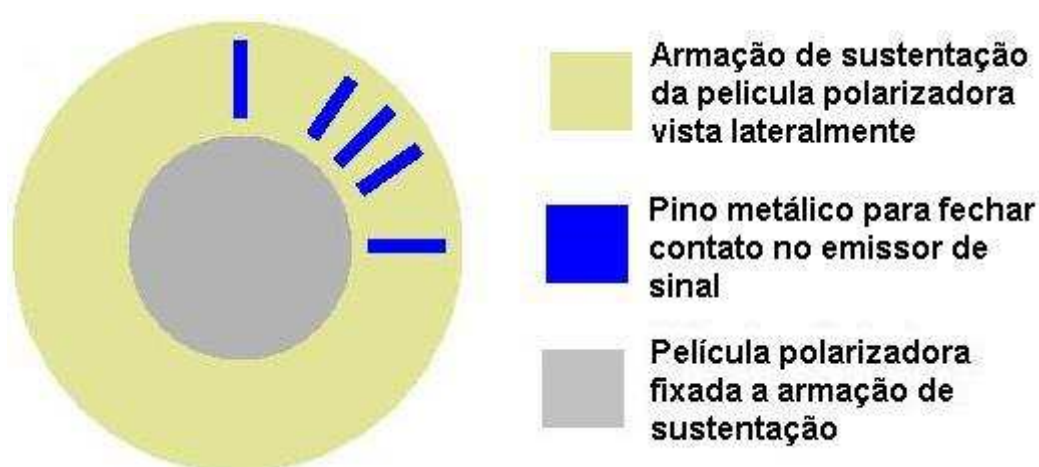


Figura 3-8 – Representação da armação de sustentação da película móvel com os metálicos de contato³⁶

3.1.3.2 Gravação do Software no Microcontrolador AT89S52

Para efetuar esta gravação foi utilizado um esquema onde o microcontrolador é ligado diretamente à porta paralela do computador pessoal. A figura 3-9 mostra as ligações entre o microcontrolador e a porta paralela do computador pessoal.

Segundo Labien (2009) as ligações feitas são:

- Porta P1.4 do microcontrolador; é ligada ao aterramento do circuito;

³⁶ Fonte: Braga (2009)

- Porta MOSI do microcontrolador; é ligada ao pino 1 da porta paralela;
- Porta MISO do microcontrolador; é ligada ao pino 11 da porta paralela;
- Porta SCK do microcontrolador; é ligada ao pino 14 da porta paralela;
- Porta RST do microcontrolador; é ligada ao pino 17 da porta paralela passando por um resistor de 1k Ω ;
- Porta VCC do microcontrolador; é ligada a uma alimentação de 5V;
- Porta GNG do microcontrolador; é ligada no aterramento.

A figura 3-9 mostra a ordem de ligação entre os pinos da porta paralela e os pinos do microcontrolador 8051.

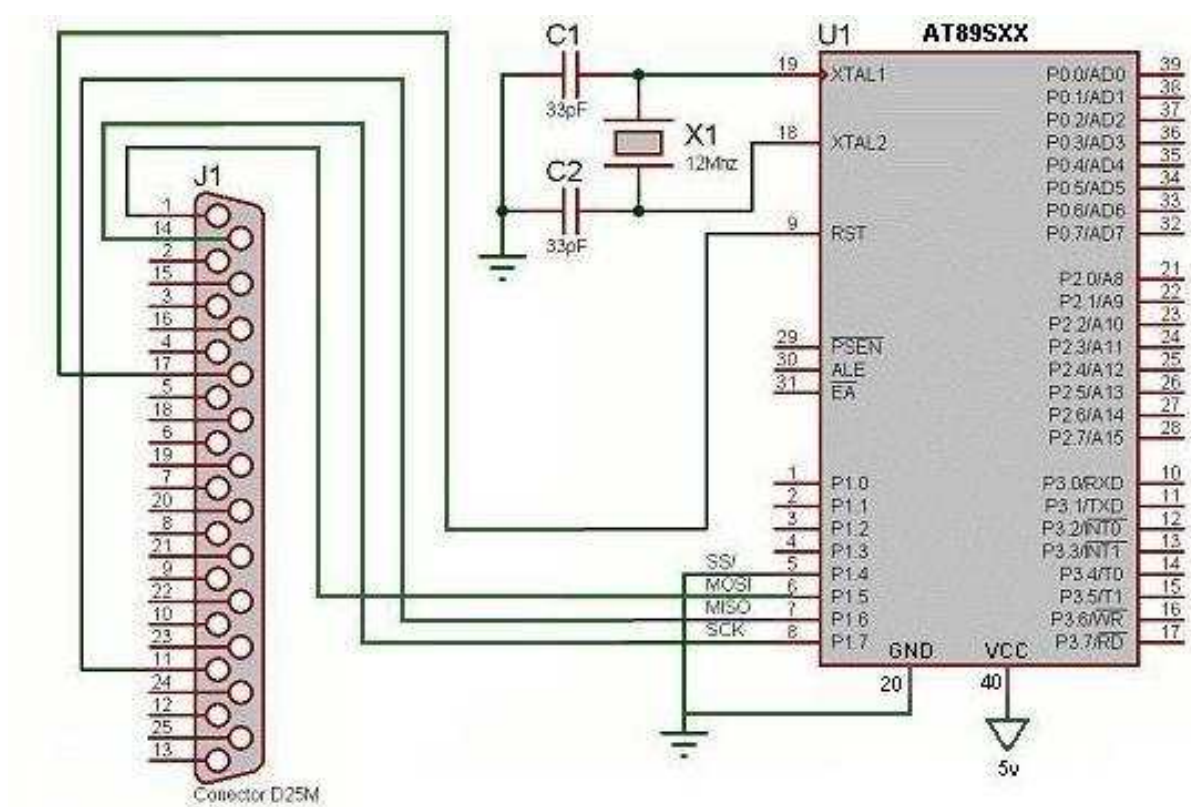


Figura 3-9 – Ligações entre microcontrolador e porta paralela para gravação do software³⁷

³⁷ Fonte: Braga (2009)

Foi utilizado o software Grava211, disponível gratuitamente no site do seu desenvolvedor “Rodrigo Ferrari”, para gravação do arquivo compilado em .HEX (formato aceito pelo microcontrolador) no microcontrolador AT89S52. Figura 3-10 mostra interface do software, (FERRARI, 2006).



Figura 3-10 – Interface do software Grava211³⁸

3.1.4 Pinagens do *Display* LCD 16X2

O *display* de LCD (*Liquid Cristal Display*) possui dezesseis pinos sendo que oito deles são para o tráfego dos dados que são escritos, dois para alimentação, um para o controle do contraste, um para habilitar o recebimento de sinais, um para informar ao *display* se nele será feito uma leitura ou escrita de dados.

A seguir segue tabela 3-2 informando as funções de cada “Porta” do *display*.

³⁸ Fonte: <http://rafaelbf.tripod.com/grava.htm>. Acessado 03 de maio de 2009

Tabela 3-2 – Especificações e siglas da pinagem do *display*³⁹

Pin No.	Symbol	Function
1	Vss	Ground
2	Vdd	+5V
3	Vo	LCD contrast adjust
4	RS	Register select
5	R/W	Read / write
6	E	Enable
7	DB0	Data bit 0
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7
+	BL+	Power Supply for BL+
-	BL-	Power Supply for BL-

Para que o *display* reconheça os dados que estão sendo enviados pelo microcontrolador é necessário enviar alguns comandos para interpretação dos dados. Na tabela 3-3 estão os comandos de configuração utilizados para escrita no *display*.

Tabela 3-3 – Comando utilizados para configuração do *display*⁴⁰

Comando	Descrição do comando
38H	Informa ao <i>display</i> que ele possui 2 linhas e 16 colunas e cada cursor é composto por uma resolução de 5X8 pixels
C0H	Avisa ao <i>display</i> para ele começar a escrever na linha 2, coluna 1
80H	Avisa ao <i>display</i> para ele começar a escrever na linha 1, coluna 1
0EH	Aciona cursor
06H	Habilita escrita da esquerda para direita
01H	Limpa o <i>display</i>

A figura 3-11 mostra a imagem do *display*. A contagem correta dos contatos do *display* se dá com a tela virada para baixo, por este motivo a contagem nesta visão será de forma decrescente começando do pino 14.

³⁹ Fonte: Datasheet do *Display LCD*

⁴⁰ Fonte: Braga (2009)

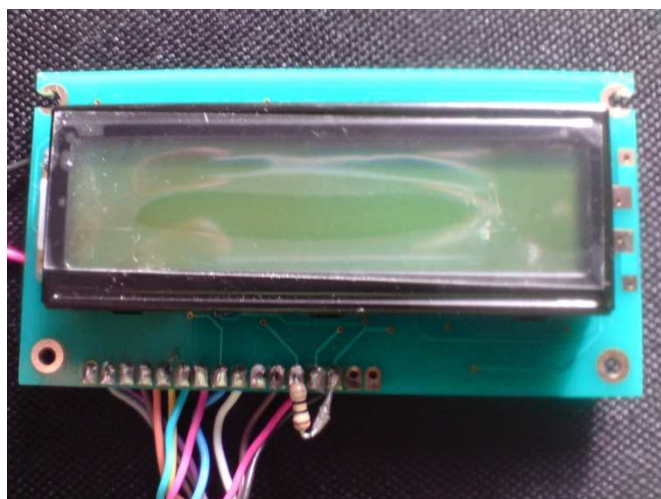


Figura 3-11 – Imagem do *display* LCD 16X2⁴¹

As ligações feitas ao microcontrolador são as seguintes:

- Pinos 7 ao 14: são responsáveis pelo recebimentos dos dados que são escritos no microcontrolador, e estão ligados nas Portas P1.0 ao P1.7 do microcontrolador, respectivamente;
- Pino 6: recebe sinais TTL para habilitar ou desabilitar o recebimento de sinais. Este pino é ligado à Porta P3.6 do microcontrolador;
- Pino 5: tem a função de informar ao *display* se é enviado um dado ou se é buscado um dado, como o projeto não utilizada a busca de dados então este pino foi ligado ao aterramento do dispositivo (a ligação não esta visível, pois se encontra na parte virada para baixo);
- Pino 4: é responsável por informar ao *display* se a ele é enviado um dado ou uma configuração, enviando o sinal TTL. Nele é habilitado à recepção de dados que aparece no *display*. Se o sinal TTL não for enviado, ele se prepara para receber uma configuração. Este pino foi ligado a Porta P3.7 do microcontrolador;

⁴¹ Fonte: Braga (2009)

- Pino 3: é um controlador de contraste do *display*, como não será usado um potenciômetro para efetuar este controle, será fixado uma intensidade de contraste através de uma ligação deste pino com o aterramento através de um resistor de 10Ω . Quanto maior a resistividade menor será o contraste;
- Pino 2: Alimentação do *display*. A alimentação deste *display* utiliza a tensão que varia de 4,5V a 5V. Como o circuito é alimentado por uma tensão de 5.5 Volts deve-se utilizar um resistor de 470Ω para efetuar esta queda de tensão;
- Pino 1: Aterramento do *display*.

Nota-se outros dois contatos após o Pino 1, que são referentes ao *backlight*, mas o modelo de LCD utilizado não veio equipado com este dispositivo.

A queda de tensão referente a ligação do Pino 2 é calculada conforme Lei de Ohm que diz:

$$V = R \times I \quad (3-1)^{42}$$

onde V é a tensão do sistema, R é a resistência e I a corrente do sistema, (UNB, 1999)

Através desta lei é calculado o resistor necessário. Como que a corrente utilizada pelo display é de 2.1 mA e a tensão de alimentação do sistema é de 5.5V. É encontrado:

$$R = \frac{5.5 - 4.5}{0.0021} = 476\Omega \quad (3-2)$$

⁴² Fonte: <http://www.unb.br/iq/kleber/EaD/Eletromagnetismo/LeiOhm/LeideOhm.html>. Acessado: 03 de julho de 2009

Devido disponibilidade do material no mercado e por apresentar uma diferença desprezível, utilizou-se um resistor de 470Ω.

3.1.5 Hardware Finalizado

Este capítulo tem a intenção de apresentar o esquema de ligações do circuito simulado no *software* “Proteus desing suite 7”, que se mostra como uma ótima ferramenta para a simulação do circuito, teste do *software* e apresentação através de imagem do *hardware* finalizado.

A seguir a figura 3-12 mostra o esquema das ligações efetuadas no circuito do projeto.

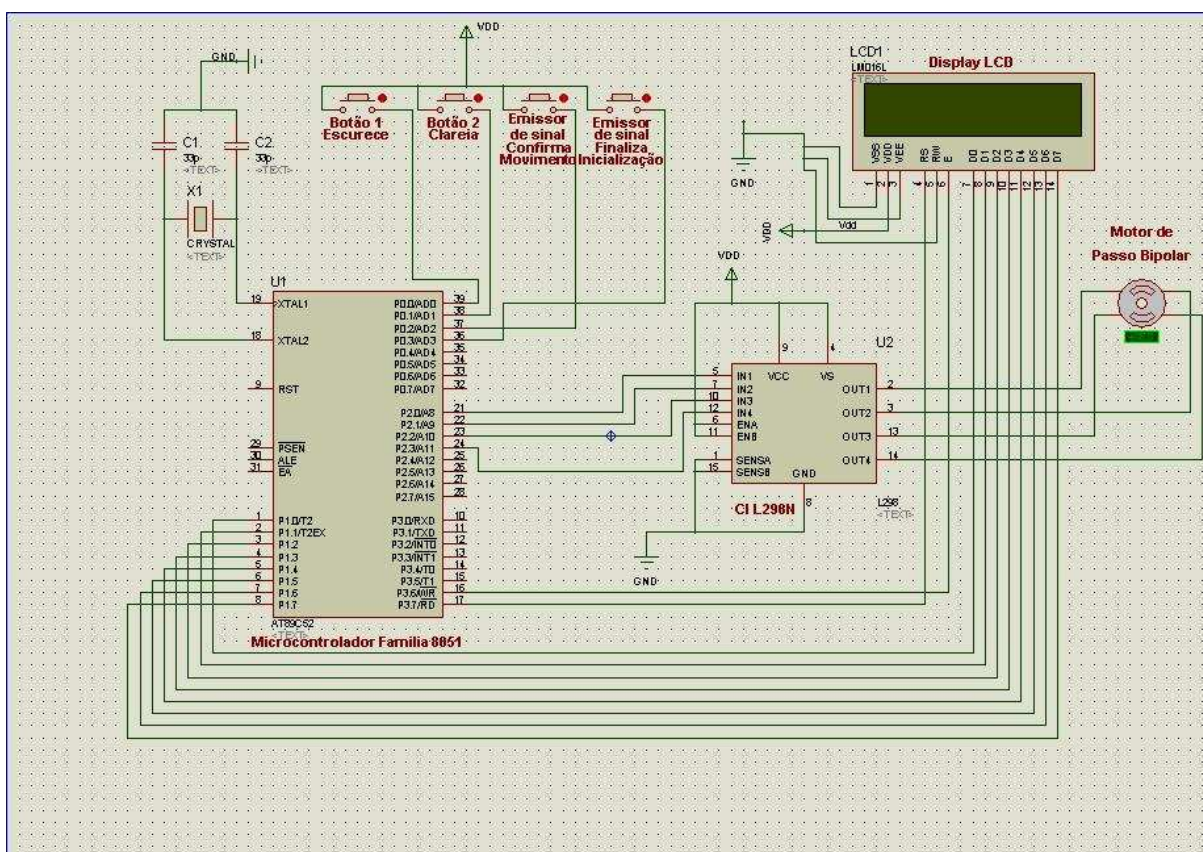


Figura 3-12 – Esquema de ligações do *hardware*⁴³

⁴³ Fonte: Braga (2009)

Já a figura 3-13, abaixo, mostra o *hardware*, no *protoboard*, desenvolvido e pronto para ser instalado na maquete.

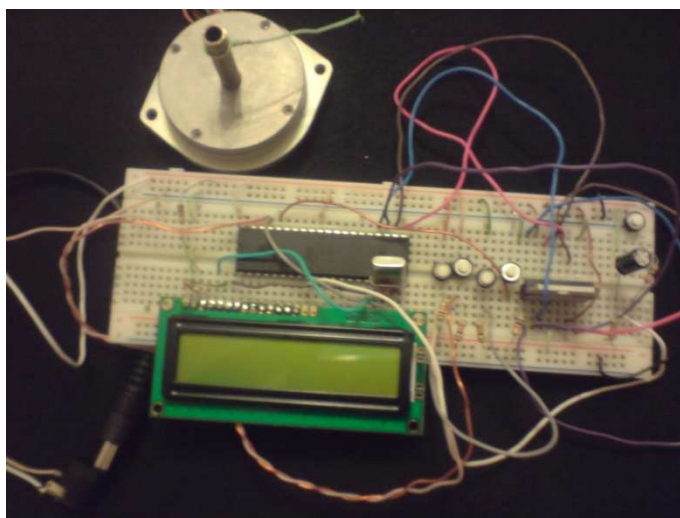


Figura 3-13 – *Hardware* finalizado⁴⁴

3.2 Desenvolvimento do *Software*

Este capítulo tem a finalidade de esclarecer a funcionalidade do *software* desenvolvido explicando as funções de algumas rotinas contidas nele.

3.2.1 Linguagens Utilizadas para a Criação do *Software*

Para o desenvolvimento deste *software* utilizou-se dois tipos de linguagens de programação. A maior parte do programa foi desenvolvida em C por ser uma linguagem de alto nível aceita pelos microcontroladores família 8051 e um pequeno trecho do envio de dados ao display de LCD foi escrito em *Assembly*.

Parte do *software* responsável pela escrita no LCD é baseado em exercício, passado pelo Prof. Eng. Anderson Royes Terroso do Departamento de Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul – *PUCRS*, sendo necessárias algumas adaptações e correções contidas no exercício, (PUCRS, 2009).

⁴⁴ Fonte: Braga (2009)

3.2.2 Fluxograma

A figura 3-14 mostra o fluxograma do *software* desenvolvido. Este fluxograma apresenta através de símbolos a rotina de seu funcionamento.

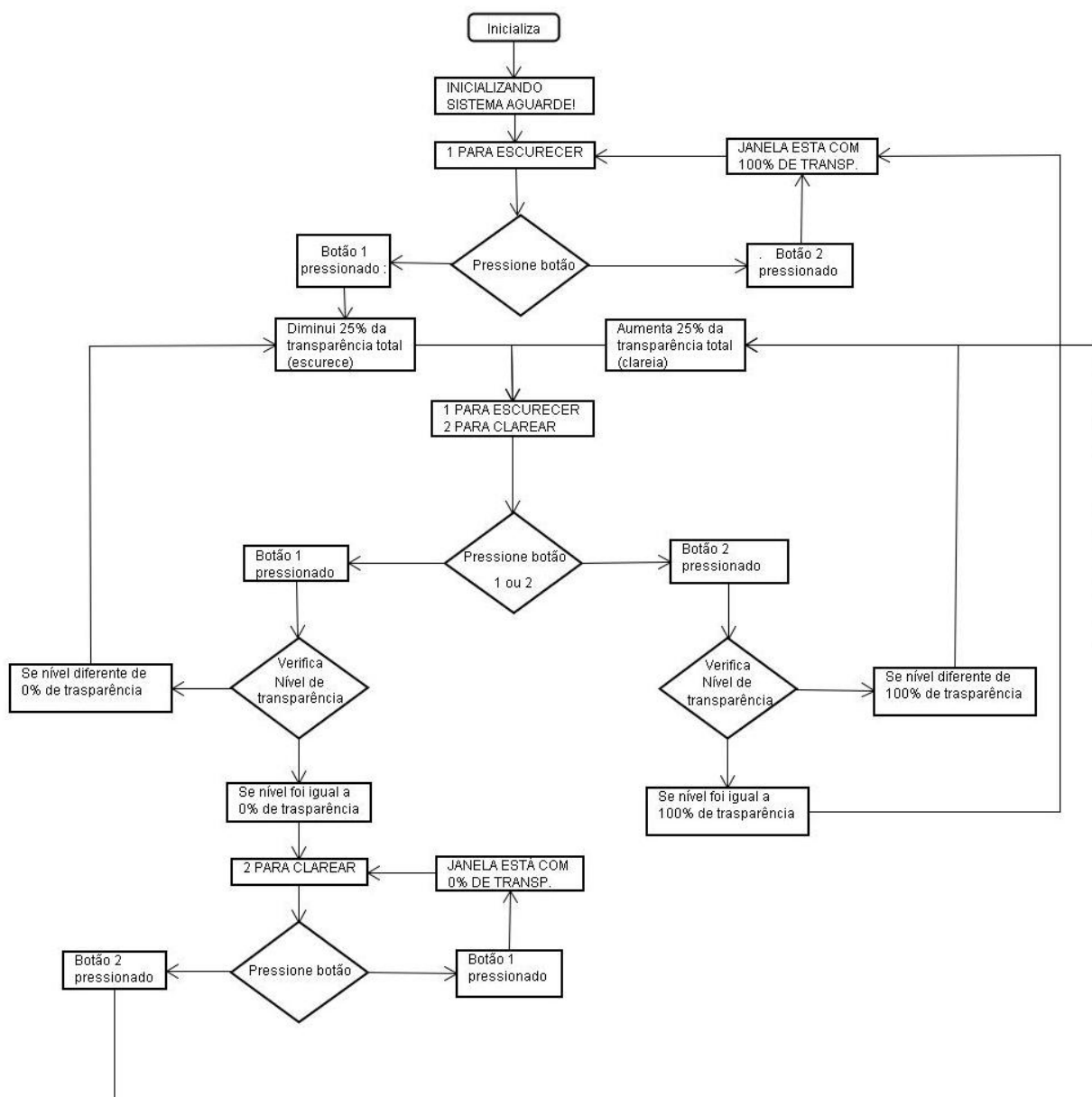


Figura 3-14 – Fluxograma do *software* desenvolvido⁴⁵

⁴⁵ Fonte: Braga (2009)

3.2.3 Funcionamento do *Software*

O *software* é a parte do projeto onde é feito a configuração de todas as tarefas que o microcontrolador deve executar. O microcontrolador e o *software* dependem um do outro neste projeto, sendo que o microcontrolador não trabalha sem o *software*.

O *software* é composto por algumas funções executadas de acordo com as requisições. Através da explicação de como o *software* funciona automaticamente é explicado todo o funcionamento do projeto.

A primeira função do *software* é a de carregar as configurações do *display*, enviando ao dispositivo informações como endereço inicial de escrita da linha 1 e linha 2, sua resolução, em qual coluna deve ser inicializada a escrita, em qual sentido são escritas as informações e limpeza dos caracteres antigos. Parte do código referente a estas configurações, é dado por:

```
void config_lcd(void)
{

    dado = 0x38;
    wr_cmd();
    dado = 0x07;
    wr_cmd();
    dado = 0x0E;
    wr_cmd();
    dado = 0x06;
    wr_cmd();
    dado = 0x01;
    wr_cmd();

}
```

Após ter carregado as configurações do *display*, é enviado um comando para que a janela gire de forma total até que seja enviado o sinal pelo dispositivo “Emissor de Sinal Finaliza Inicialização”, conforme figura 3-12. Esta parte do programa é referente ao sistema contra falhas, citado anteriormente. O trecho do código responsável pela execução desta tarefa, é dado por:


```

    }
}

```

Finalizada a operação de inicialização, o microcontrolador começa a parte de interatividade com o usuário onde ele pode escolher entre os níveis de transparência pré-determinados informados no *display*, as opções que ele tem, “1 para escurecer” e “2 para clarear”, onde “1” e “2” são os números referentes aos botões disponíveis ao usuário: o botão “1” (preto) reduz o nível de transparência da janela e o botão “2” (vermelho) aumenta o nível de transparência. O trecho do código referente a esta operação, é dado por:

```

void retorna_tela_inicial(void)
{
    linha1();
    escreve_char("1 PARA ESCURECER");
    linha2();
    escreve_char("2 PARA CLAREAR ");
}

```

Estas duas opções são mostradas ao usuário, exceto quando os níveis de transparência estiverem nos limites “0%” e “100%”, neste caso o sistema mostra apenas a opção disponível para ele, se o sistema estiver em 0% de transparência (completamente escura) o sistema dá somente a opção de aumentar o nível de transparência, se estiver em 100% mostra somente a opção para diminuí-lo. O trecho do código que mostra no display a opção disponível para quando a janela estiver com nível máximo de transparência, é dado por:

```

if (est_geral == 0)
{
    linha1();
    escreve_char("1 PARA ESCURECER");
    linha2();
    escreve_char("          ");
}

```

O código da opção que é mostrada no display quando a janela estiver no nível mínimo de transparência, é dado por:


```

        break;
    case 4:
        led1=0;
        led2=0;
        led3=0;
        led4=1;
        break;
    default:
        break;
    }
    x=4;
}
w--;
}
while(bot3 == 0 )
{
    for(i=x;i>0;i--)
    {
        if(bot3 == 1)
        {
            status = i;
            break;
        }
        tempo();
        switch(i)
        {
            case 1:
                led1=1;
                led2=0;
                led3=0;
                led4=0;
                break;
            case 2:
                led1=0;
                led2=1;
                led3=0;
                led4=0;
                break;
            case 3:
                led1=0;
                led2=0;
                led3=1;
                led4=0;
                break;
            case 4:
                led1=0;
                led2=0;

```



```

        led3=0;
        led4=1;
        break;

    default:
        break;
    }
    x=1;
}
w--;
}

while(bot3 == 0)
{
    for(i=x;i<5;i++)
    {
        if (bot3 == 1)
        {
            status = i;
            break;
        }

        tempo();
        switch(i)
        {
            case 1:
                led1=1;
                led2=0;
                led3=0;
                led4=0;
                break;
            case 2:
                led1=0;
                led2=1;
                led3=0;
                led4=0;
                break;
            case 3:
                led1=0;
                led2=0;
                led3=1;
                led4=0;
                break;
            case 4:
                led1=0;
                led2=0;

```

```

        led3=0;
        led4=1;
        break;

    default:
        break;
}

}
x=1;
}
}

```

Para evitar problemas como um acionamento indevido do botão para clarear ou escurecer quando já se estiver no limite destes níveis, é incorporado ao programa um contador que gerencia em qual nível de transparência o sistema se encontra, informando ao usuário que ele já está no nível máximo permitido pela janela, assim desprezando o movimento solicitado e informando novamente qual a única opção de movimento que ele possui. O trecho responsável por esta verificação quando informado somente a opção para escurecer a janela, é dado por:

```

else
{
    linha1();
    escreve_char("JANELA ESTA COM ");
    linha2();
    escreve_char(" 100% DE TRANSP.");
    est_geral = 0;
    tempo_mensag_limite();
    linha1();
    escreve_char("1 PARA ESCURECER");
    linha2();
    escreve_char("      ");
}

```

O segue trecho responsável pela verificação quando informado somente a opção para clarear a janela, é dado por:

```

else
{
    linha1();
    escreve_char("JANELA ESTA COM ");

```

```

        linha2();
        escreve_char(" 0% DE TRANSP. ");
        est_geral=4;
        tempo_mensag_limite();
        linha1();
        escreve_char("2 PARA CLAREAR ");
        linha2();
        escreve_char("          ");
    }

```

Outra característica do *software* é informar ao usuário qual tipo de movimento a janela está realizando e para qual nível de transparência a janela está se posicionando. Caso a janela esteja efetuando um movimento de redução do nível de transparência, ela informa “ESCURECENDO P/ XX% DE TRANSP”, onde XX se refere aos níveis pré-determinados disponibilizados no sistema, informando assim o tipo de movimento em realização e para qual nível ele está indo. O mesmo ocorre quando o movimento for o contrário, neste caso será exibida a mensagem “CLAREANDO P/ XX% DE TRANSP”.

Os trechos do código referente a informação de que a janela está clareando, são dados por:

```

else
    transp--;
    linha1();
    escreve_char("CLAREANDO P/ ");
    linha2();
    mostra_transp();

```

Os trechos do código referente a informação de que a janela está escurecendo, são dados por:

```

else
    transp++;
    linha1();
    escreve_char(" ESCURECENDO P/ ");
    linha2();
    mostra_transp();

```

A ultima linha dos últimos dois trechos de código apresentado, que se refere ao código “mostra_transp();”, está chamando uma função que é responsável por dizer para qual o nível de transparência a janela está seguindo. A função que faz este controle, é dada por:

```
void mostra_transp(void)
{
    switch(transp)
    {
        case 0:
            escreve_char("100% TRANSP.  ");
            break;

        case 1:
            escreve_char("75% TRANSP.  ");
            break;
        case 2:
            escreve_char("50% TRANSP.  ");
            break;
        case 3:
            escreve_char("25% TRANSP.  ");
            break;
        case 4:
            escreve_char("0% TRANSP.  ");
            break;
    }
}
```

O código fonte do programa comentado é dado no Apêndice desta monografia.

4 RESULTADOS OBTIDOS

Do ponto de vista de hardware, o projeto possui dois *push buttons* para interatividade com usuário e dois emissores de sinais utilizados como sensores de interrupção do movimento, informando ao sistema que a janela atingiu o nível de transparência desejado.

4.1 Mensagens de Interação com o Usuário

O projeto é configurado com um sistema para eliminar possíveis falhas quando existir a ausência de energia elétrica. Este sistema faz-se necessário durante a inicialização, pois o motor de passo perde seu torque quando ocorre este evento, o que o deixa vulnerável a movimentos causados por qualquer natureza.

Este sistema tem a função de toda vez que o microcontrolador for energizado girar totalmente a janela de forma a deixá-la com seu nível máximo de transparência, isso ocorre quando o dispositivo batizado de “Emissor de Sinal Finaliza Inicialização” for acionado. Quando este dispositivo for acionado o *software* “saberá” que a janela está em seu nível máximo de transparência, assim podendo inicializar sua interatividade com o usuário oferecendo a precisão dos níveis de transparência.

A figura 4-1 mostra o sistema nesta fase de inicialização.

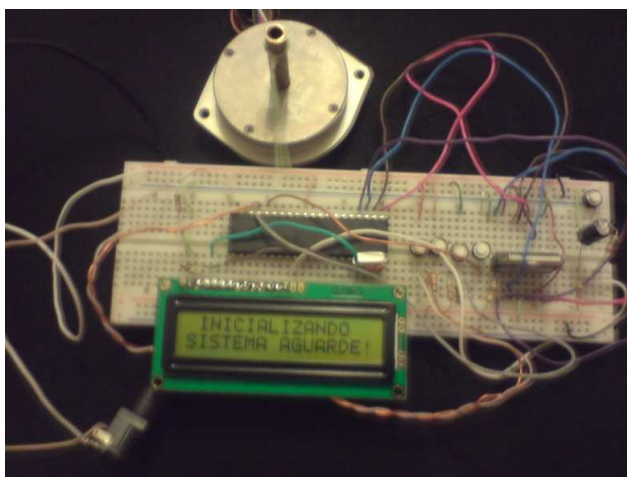


Figura 4-1 – Demonstração da tela de inicialização⁴⁶

⁴⁶ Fonte: Braga (2009)

Ao final da a fase de inicialização aparece no *display* a informação das opções que o usuário tem para aquele momento, se a janela estiver no nível máximo de transparência ela pode somente escurecer (figura 4-2) e se estiver no nível mínimo de transparência ela pode somente clarear (figura 4-3).

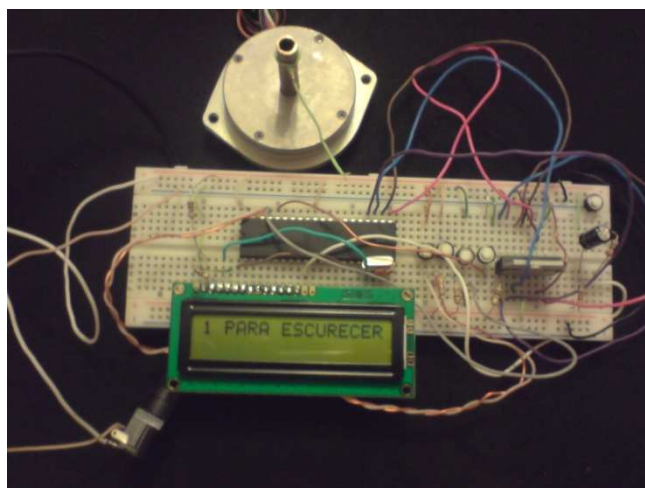


Figura 4-2 – Sistema disponibilizando apenas opção para escurecer⁴⁷

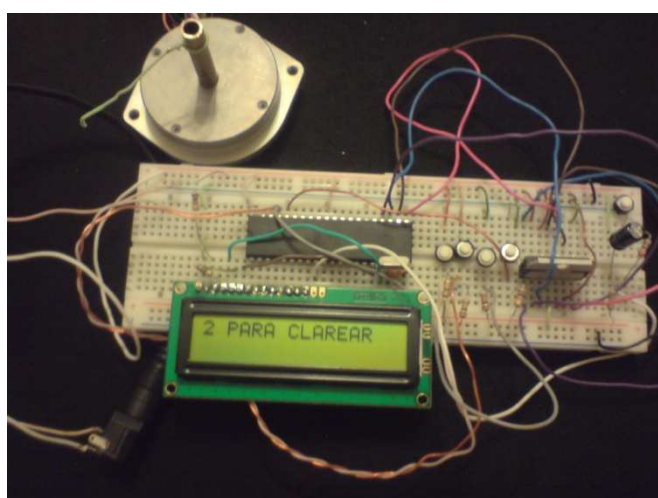


Figura 4-3 – Sistema disponibilizando apenas opção para clarear⁴⁸

Quando a janela não estiver em um de seus níveis limites de transparência (máximo e mínimo) o usuário tem a escolha entre as duas ações, sendo no *display* informado suas opções, conforme figura 4-4.

⁴⁷ Fonte: Braga (2009)

⁴⁸ Fonte: Braga (2009)

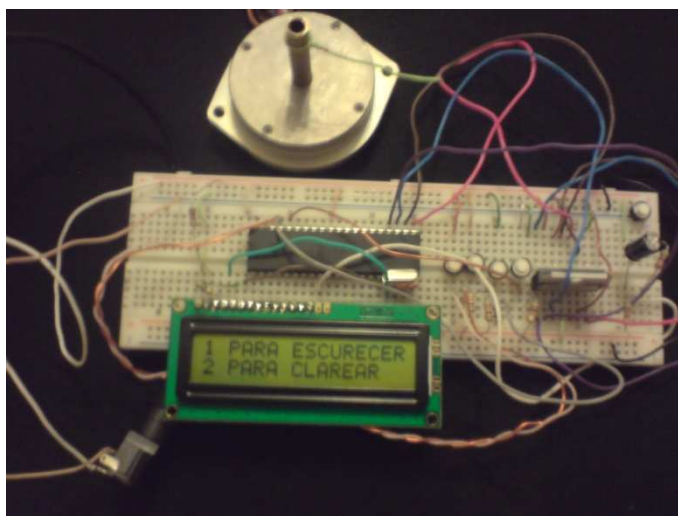


Figura 4-4 – Sistema disponibilizando duas opções de movimento⁴⁹

Para que o usuário tenha idéia do que está ocorrendo no sistema durante um determinado movimento, o sistema informa a ele qual o tipo de ação que está sendo executada e para qual nível de transparência a janela está sendo configurada. Quando for dado o comando para escurecer a janela, no LCD aparece a mensagem “ESCURECENDO P/ XX% TRANSP”, desta maneira informando ao usuário que a janela esta escurecendo e indo para o nível XX de transparência, conforme a figura 4-5.

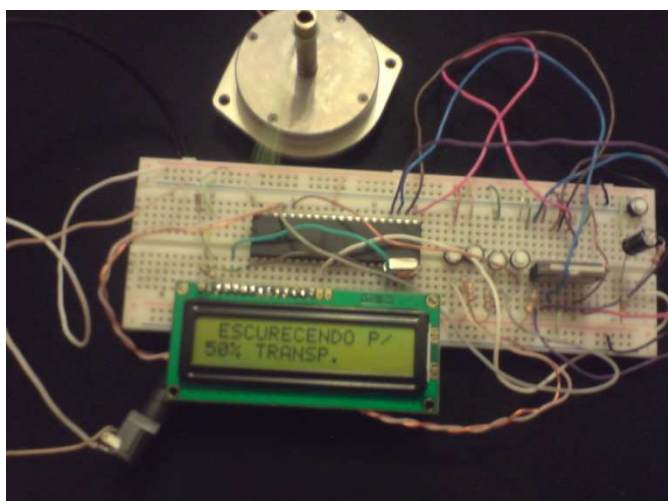


Figura 4-5 – Sistema informando que a janela está escurecendo e caminhando para 50% da transparência total⁵⁰

⁴⁹ Fonte: Braga (2009)

⁵⁰ Fonte: Braga (2009)

A figura 4-6 mostra o sistema informando o contrário da figura 4-5, que está diminuindo o nível de transparência. Durante esta ação, o *display* informa “CLAREANDO P/ XX% TRANSP”, passando a informação ao usuário de que a janela está aumentando o nível de transparência e indo para o valor XX.

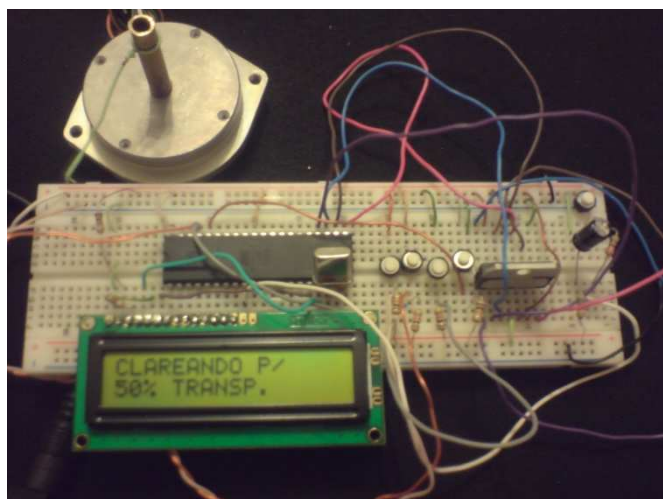


Figura 4-6 – Sistema informando que a janela está clareando e caminhando para 50% da transparência total⁵¹

O sistema desenvolvido também tem a função de advertir o usuário caso ele tente ultrapassar os níveis pré-determinados. Caso o dispositivo já esteja no seu nível máximo de transparência for pressionado o *push button* para clarear, aparece no LCD a mensagem “JANELA ESTA COM 100% DE TRANSP”, conforme a figura 4-7, e posteriormente o sistema voltar a dar a única opção que o usuário possui neste momento, que é a de escurecer a janela.

⁵¹ Fonte: Braga (2009)

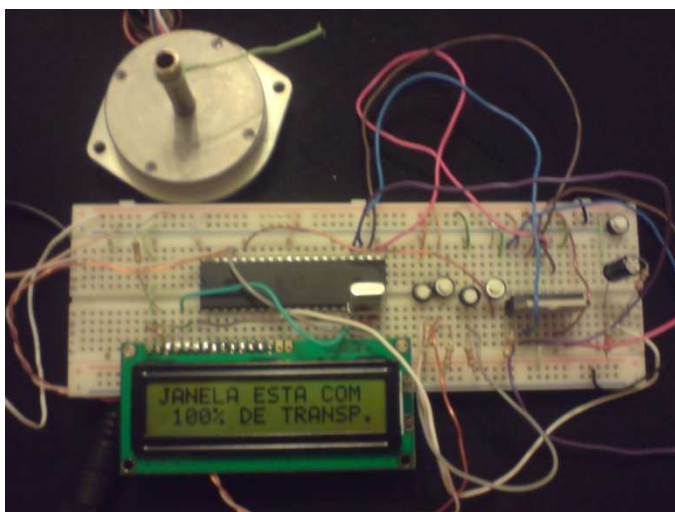


Figura 4-7 – Imagem mostrada ao pressionar *push button* indevido quando janela estiver com nível máximo de transparência.⁵²

O mesmo ocorre quando for pressionado o *push button* para escurecer quando o nível de transparência já estiver em 0% de transparência, é escrito no *display* “JANELA ESTA COM 0% DE TRANSP” conforme figura 4-8.

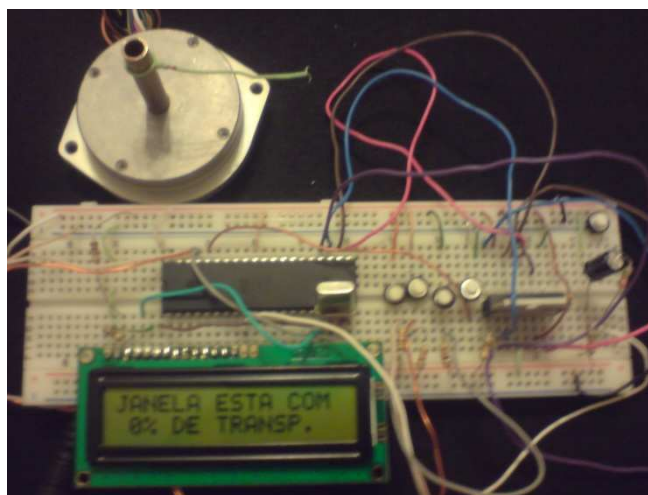


Figura 4-8 – Imagem mostrada ao pressionar *push button* indevido quando janela estiver com nível mínimo de transparência.⁵³

⁵² Fonte: Braga (2009)

⁵³ Fonte: Braga (2009)

4.2 Montagem da Maquete

Para demonstração à Banca Examinadora foi confeccionada uma maquete para simular um cômodo residencial, onde foi acoplado o *hardware* para o controle da janela.

Na maquete as duas películas polarizadoras foram colocadas uma fixa e a outra móvel, para que possa se efetuar o movimento necessário para se obter o efeito da polarização desejado.

As películas polarizadoras possuem uma armação de madeira para mantê-las esticadas e dar rigidez as suas margens, sendo que a película móvel recebe uma armação com o diâmetro externo maior.

A película móvel com a armação de diâmetro externo maior é presa à janela através de roldanas, que firmam esta película para que ela não caia. As roldanas possuem também a função de trilhos para que a armação da película não perca o atrito com o motor de passo.

O sistema que faz a armação da película móvel girar é composto por um anel de madeira preso ao eixo do motor de passo, com um material emborrachado colado em sua volta para dar o atrito necessário para girar a película móvel.

Abaixo seguem as figuras 4-9 e 4-10 que representam o esquema da janela.

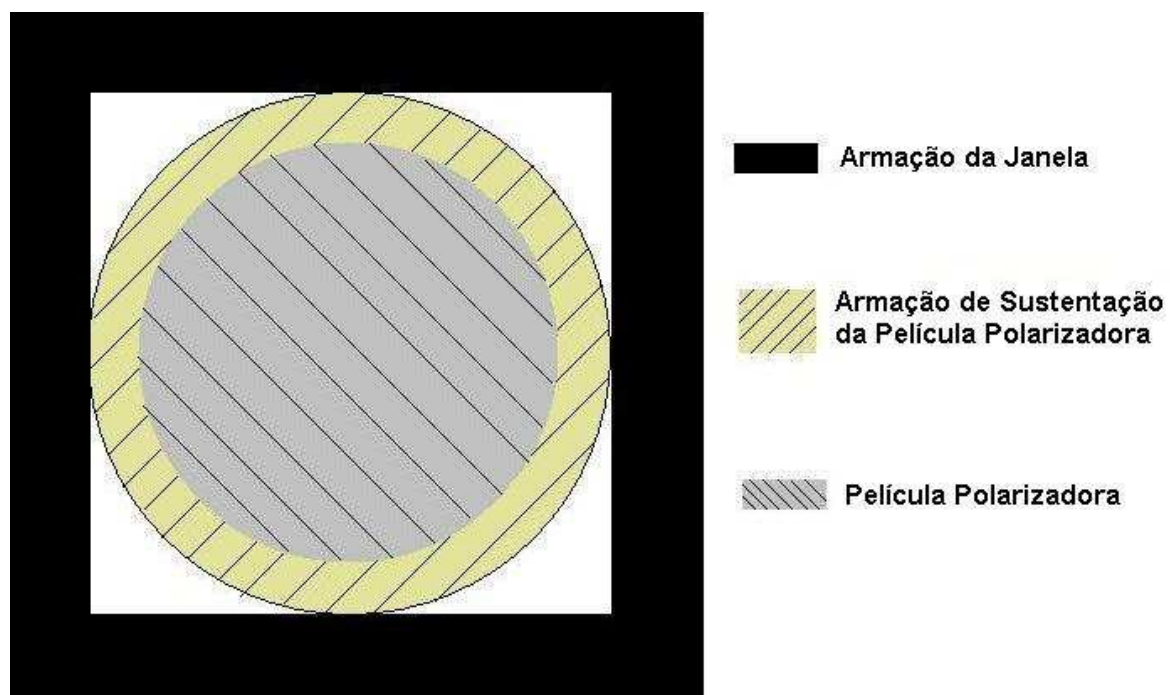


Figura 4-9 – Esquema da Película fixa⁵⁴

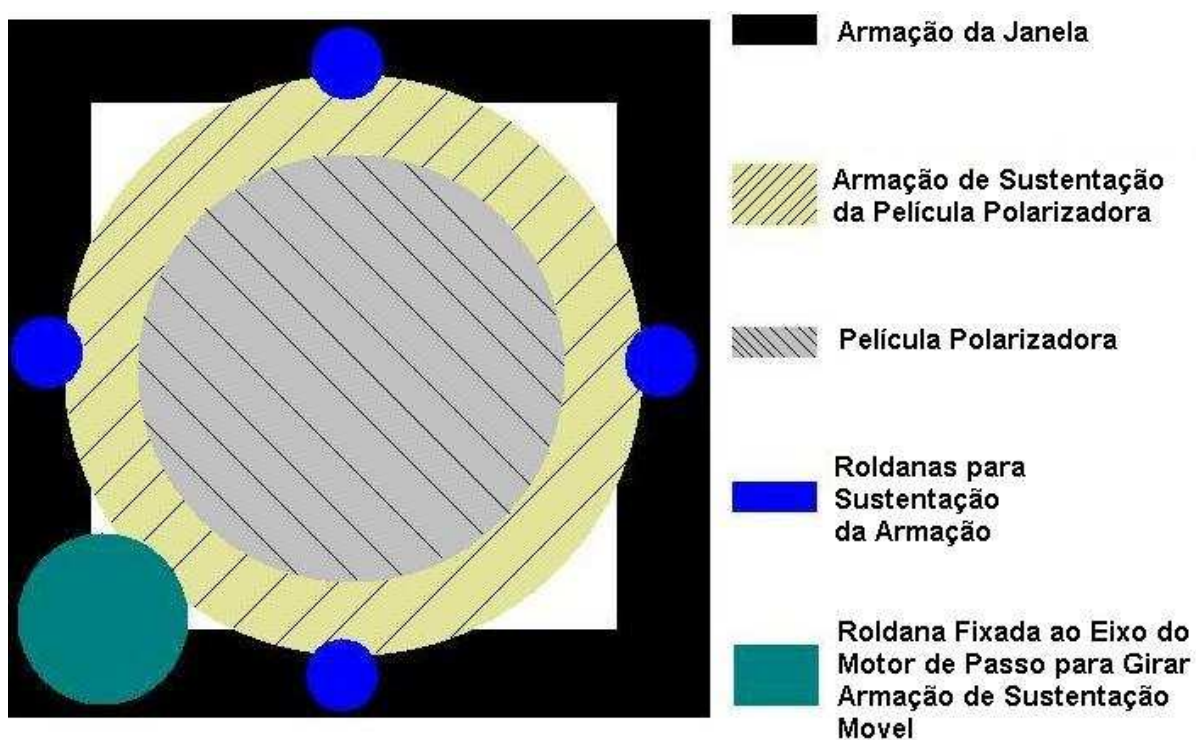


Figura 4-10 – Esquema da Película móvel⁵⁵

⁵⁴ Fonte: Braga (2009)

⁵⁵ Fonte: Braga (2009)

Para que a janela atinja os níveis de transparência disponibilizados no sistema, a janela precisa formar determinadas diferenças angulares em relação ao sentido da polarização de suas películas.

As diferenças angulares dos sentidos das polarizações das películas são calculados através da Lei de Malus, informada na equação 2-1 desta monografia, e estão determinadas na tabela 4-1 que mostra a relação nível de transparência e ângulo entre sentidos das polarizações da películas.

Para marcação das diferenças angulares nas armações de sustentação das películas foi utilizado o seguinte esquema: primeiro foram colocadas as películas paralelamente (uma sobre a outra). Após este processo uma película foi sendo girada até notar que a luz já não atravessava o sistema e que girando mais um pouco a luz voltava a atravessá-lo. Achado este ponto foi feito na película uma pequena marcação para não perder o nível de 0% de transparência e com o auxílio de um transferidor foi marcado os ângulos encontrados pela lei de Malus conforme tabela 4-1.

Tabela 4-1 – Relação entre nível de transparência e ângulo entre películas⁵⁶

Nível de transparência	Ângulo entre Películas
100%	0°
75%	30°
50%	45°
25%	60°
0%	90°

4.2.1 Imagens da Maquete

Este sub-capítulo é destinado a ilustrar as imagens da maquete confeccionada para apresentação a banca examinadora. A seguir são apresentadas as imagens.

⁵⁶ Fonte: Braga (2009)

A figura 4-11 mostra a armação de sustentação das películas polarizadoras e motor de passo, com uma visão da parte frontal.

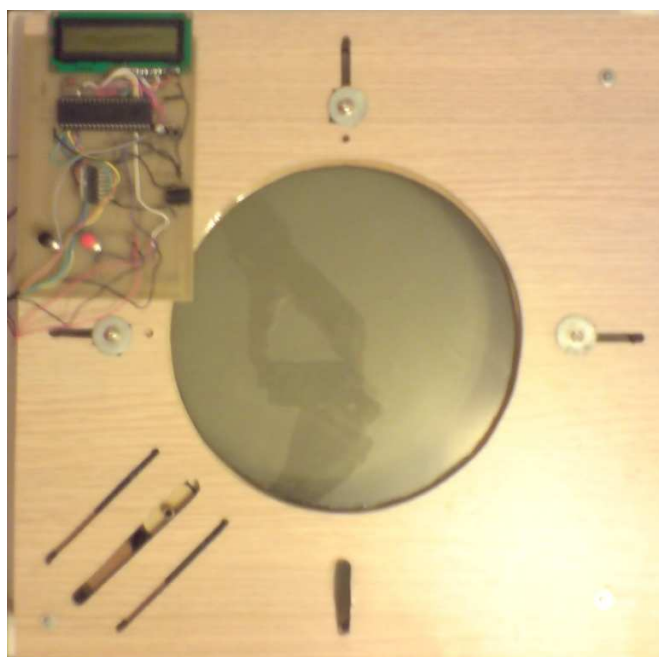


Figura 4-11 – Armação para sustentação dos componentes vista pela frente⁵⁷

A figura 4-12 mostra a armação de sustentação, das películas polarizadoras e motor de passo, com uma visão da parte de trás.

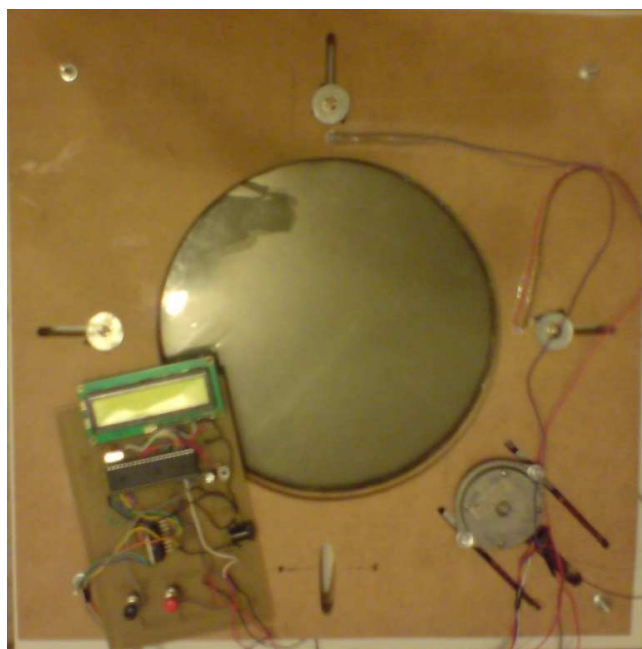


Figura 4-12 – Armação para sustentação dos componentes vista por trás⁵⁸

⁵⁷ Fonte: Braga (2009)

⁵⁸ Fonte: Braga (2009)

Na figura 4-12, pode-se visualizar o motor de passo fixado e mais dois pares de fios saindo da armação. Os dois pares de fios na parte superior da imagem são responsáveis pela emissão do sinal que finaliza a inicialização do sistema (sistema contra falhas) e o outro par é responsável por emitir o sinal que confirma o movimento de mudança de nível de transparência solicitado pelo usuário realizado com sucesso.

A figura 4-13 mostra a parte entre as duas placas de sustentação, onde é possível visualizar a película móvel que está com uma armação escura, as roldanas para sustentação e guia da película móvel e uma pequena parte da armação da película fixa com uma cor clara sob a armação da película móvel.

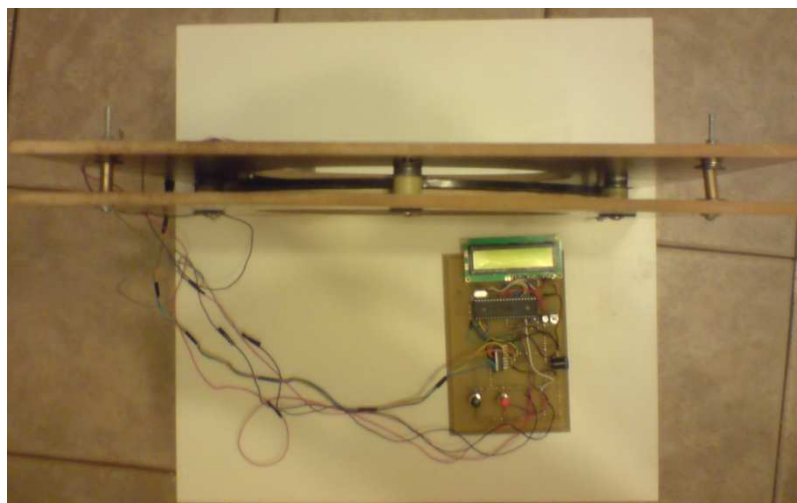


Figura 4-13 – Armação para sustentação dos componentes mostrando a montagem entre as placas⁵⁹

A figura 4-14 mostra a caixa onde são encaixadas as armações para que a maquete possa simular um cômodo. Esta caixa possui um tamanho de aproximadamente 40cm³ de volume.

⁵⁹ Fonte: Braga (2009)



Figura 4-14 – Caixa para encaixe das armações de sustentação dos componentes⁶⁰

A figura 4-15 mostra a caixa já com as armações encaixadas.



Figura 4-15 – Caixa com armações de sustentação acopladas⁶¹

A figura 4-16 mostra a maquete pronta e funcionando em seu estado máximo de transparência, ou seja, 100% de transparência permitida pelo sistema.

⁶⁰ Fonte: Braga (2009)

⁶¹ Fonte: Braga (2009)



Figura 4-16 – Maquete pronta mostrando películas com nível máximo de transparência⁶²

A figura 4-17 mostra a maquete com 75% do nível total de transparência permitido pelo sistema.



Figura 4-17 – Maquete pronta mostrando películas com 75% de transparência⁶³

⁶² Fonte: Braga (2009)

⁶³ Fonte: Braga (2009)

Na figura 4-18 mostra a maquete com 50% do nível total de transparência permitido pelo projeto.



Figura 4-18 – Maquete pronta mostrando películas com 50% de transparência⁶⁴

A figura 4-19 mostra maquete com 25% do nível total disponível pelas películas.



Figura 4-19– Maquete pronta mostrando películas com 25% de transparência⁶⁵

A figura 4-20 mostra o projeto com o nível mínimo de transparência.

⁶⁴ Fonte: Braga (2009)

⁶⁵ Fonte: Braga (2009)



Figura 4-20– Maquete pronta mostrando películas com 0% de transparência⁶⁶

⁶⁶Fonte: Braga (2009)

5 CONCLUSÃO

O sistema desenvolvido e testado neste projeto acadêmico um sistema para o controle de luminosidade solar em ambientes, exigiu a implementação tanto de *software* quanto de *hardware* para que pudesse ter o controle sobre a janela.

Este projeto trouxe um ganho significativo de conhecimento sobre os materiais utilizados para confecção do *hardware* como: microcontrolador AT89S52, CI L289N , motor de passo, *display* LCD 16X2, *push buttons*, resistores, capacitores e nas linguagens de programação C e *Assembly* que foram utilizadas.

O sistema inicializou configurando a janela com 100% de transparência, após este processo foi iniciado com o usuário a interatividade para seu controle. Com um funcionamento constante (com a existência de energia elétrica) o sistema está sempre pronto para receber uma nova opção de comando disponível, que está sendo informado através do *display* de LCD. Depois de recebido o novo comando o motor de passo gira a película móvel até janela atingir o próximo nível pré-programado deixando a opção para o usuário fazer uma nova escolha de movimento.

O projeto apresentou algumas dificuldades como a utilização do protoboard com constantes maus contatos e dificuldade em conexão dos fios e o encontro das películas polarizadoras, que não foram encontradas no mercado para aquisição, sendo necessário retirar este material de monitores de LCD sucateados o que gerou um trabalho significativo, pois foi necessário retirar a cola que a fixava ao equipamento. Para tentativa de resolução do problema da aquisição da película foram feitos contatos com fabricantes de monitores LCD como Gradiente e Philips que responderam não possuir este material.

O resultado final do projeto foi satisfatório alcançando os resultados esperados no controle de luminosidade. As Películas polarizadoras funcionaram como um ótimo material para este tipo de controle, podendo trazer para seus usuários benefícios como qualidade de vida, privacidade e conforto.

Desta forma o projeto mostrou-se bastante satisfatório no seu propósito funcionando como um sistema eficiente para o controle do nível de luz solar nos ambientes, possuindo um *hardware* de simples implementação e baixo custo.

A sugestão para trabalhos futuros baseia-se no controle do sistema através da quantidade de luz no interior do ambiente (através da quantidade de lumens, que é uma unidade de medida utilizada para medir a intensidade luminosa) e não através da escolha do nível de transparência das películas polarizadoras como apresentado neste projeto, assim sendo possível manter um nível de iluminação solar fixa dentro do ambiente independente das variações de intensidade de luz solar.

REFERÊNCIAS BIBLIOGRAFICAS E SITES CONSULTADOS

JACOB, M., HALKIAS, C. **“Eletrônica: dispositivos e Circuitos”**. 4ª Edição. São Paulo: Editora McGraw-Hill do Brasil.

MARINHO, EDINALDO DOS SANTOS; MARINHO, JOSE EDSON DOS SANTOS. **“Mini curso de Microcontrolador”**. Saber Eletrônica Especia nº2, Janeiro 2001.

MENDONÇA, ALEXANDRE; ZELENOVSKY RICARDO. **“Microcontroladores. Programação e projeto com a família 8051”**. Editora MZ Editora, 2005.

NICOLOSI, DENYS E. C. **“Laboratório de Microcontroladores Família 8051”** 5ª Edição. Editora Érica, 2008.

NICOLOSI, DENYS E. C.. **Microcontrolador 8051 Detalhado**. São Paulo: Érica, 2000.

SÁ, MAURICIO CARDOSO DE. **“Programação C para Microcontroladores 8051”**. Editora Érica.

WATERS, F. J., VALENTE, R. B. **“ABC dos componentes eletrônicos”**. São Paulo: Editora Antenna, 1979.

DA-RIN, BENEDITO PIROPO. Disponível em:
<http://www.bpiropo.com.br/cf20050718.htm>. Acessado: 01 de março de 2009.

EBAH. Disponível em: <http://www.ebah.com.br/motor-de-passo-pdf-pdf-a5479.html>. Acessado: 20 de março de 2009.

ELECTRÓNICA. Disponível em: <http://www.electronica-pt.com/index.php/content/view/139/37/>. Acessado: 20 de março de 2009.

FERRARI, RAFAEL. Disponível em: <http://rafaelbf.tripod.com/>. Acessado em 04 de abril de 2009.

IME. Disponível em:

http://www.ime.eb.br/~pinho/micro/trabalhos/Mecatronica_TP1.pdf. Acessado: 25 de março de 2009.

LABIEN. Disponível em: <http://www.labiem.cpgei.cefetpr.br>. Acessado em 04 de abril de 2009.

PUCRS. Disponível em:

http://www.inf.pucrs.br/~eduardob/disciplinas/ProgPerif/sem04.2/Seminarios/RodrigoWertonge/seminario_LCD.doc. Acessado: 15 de março de 2009.

PUCRS. Disponível em: <http://www.ee.pucrs.br/~terroso/html/lcd.html>. Acessado: 22 de março de 2009.

PUCRS. Disponível em:

http://www.ee.pucrs.br/~terroso/PROGRAMA2_C_LCD_DA_PLACA_2007_2.pdf.

Acessado: 23 de abril de 2009.

PUCRS. Disponível em: <http://fisica.fe.up.pt/luz/polarizacao.html>

Acessado em 24 de março de 2009

REVISTA BRASILEIRA DE ENSINO DE FÍSICA. Disponível em:

http://www.scielo.br/scielo.php?pid=S0102-47442002000100006&script=sci_arttext.

Acessado: 27 de abril de 2009.

SERAFIN, CLEBER. Disponível em:

<http://www.fag.edu.br/professores/cleberserafin/NUTRI%C7%C3O%20BIOQUIMICA/APOSTILA-%20BIOQU%CDMICA.doc>. Acessado em 23 de fevereiro de 2009.

UFC. Disponível em:

<http://www.seara.ufc.br/tintim/fisica/polarizacao/polarizacao4.htm>. Acessado: 24 de março de 2009.

UFF Disponível em:

<http://www.telecom.uff.br/pet/petws/downloads/tutoriais/stepmotor/stepmotor2k81119.pdf>. Acessado: 20 de março de 2009.

UFRJ. Disponível em: http://www.gta.ufrrj.br/grad/01_1/motor/index.htm. Acessado: 20 de março de 2009.

UNB. Disponível em: <http://www.unb.br/iq/kleber/EaD/Fisica-4/Aulas/Aula-17/aula-17.html>. Acessado: 24 de março de 2009.

UNB. Disponível em:

<http://www.unb.br/iq/kleber/EaD/Eletromagnetismo/LeiOhm/LeideOhm.html>.

Acessado: 03 de julho de 2009.

UNICAMP. Disponível em:

http://www.ifi.unicamp.br/~lunazzi/F530_F590_F690_F809_F895/F809/F809_sem1_2004/009632PriscilaF-Monica_F809_RF.pdf. Acessado em 15 de março de 2009.

UNICAMP. Disponível em:

<http://www.ifi.unicamp.br/~accosta/roteiros/18/nota%2018.html>. Acessado: 25 de março de 2009.

UC, disponível em:

<https://woc.uc.pt/fisica/getFile.do?tipo=2&id=699>. Acessado 29 de fevereiro de 2009.

USP. Disponível em: <http://www.cdcc.sc.usp.br/ondulatória/sinotico.html>. Acessado: 01 de março de 2009.

USP. Disponível em:

<http://fap01.if.usp.br/~vannucci/05a%20aula%20Eletromagnetismo%20II.pdf>.

Acessado em 22 de março de 2009.

APÊNDICE

```
#include "8051.h"

#define dado P1 //declarando porta P1 para envio de dados ao LCD
#define rs P3_7 //declarando os sinais P3.7 ao sinal RS
#define en P3_6 //declarando os sinais P3.6 ao sinal EN
#define led1 P2_0 // declara porta para controlar passo 1 do motor
#define led2 P2_1 // declara porta para controlar passo 2 do motor
#define led3 P2_2 // declara porta para controlar passo 3 do motor
#define led4 P2_3 // declara porta para controlar passo 4 do motor
#define bot1 P0_0 // botao aciona incrementador
#define bot2 P0_1 // botao aciona decrementador
#define bot3 P0_2 // botao interrompe incremento ou decremento
#define bot4 P0_3 // botao indica inicio de movimento(é o estado inicial)

void config_lcd(void); //função configura lcd
void linha1(void); //função escreve linha 1 do lcd
void linha2(void); //função escreve linha 2 do lcd
void escreve_char(char msn[]); //função escreve string
void delay_6us(void); //função de delay de 6 microsegundos
void delay_6ms(void); //função de delay de 8 milisegundos
void wr_cmd(void); //função para envio de configuração ao lcd
void wr_char(void); //função para envio de string ao lcd
void incrementa_led(int x); //função escurece janela
void decrementa_led(int x); //função clareia janela
void tempo(void); //função de intervalo de tempo entre passos do motor
void tempo_mensag_limite(void); //função de tempo para mostrar de mensagem
void retorna_tela_inicial(void); //função de opções 1 e 2 ao usuário
void retorna_motor(void); //função inicializa sistema
int status = 2; //inicializa sistema com status do motor no passo 2
int transp = 0; // indica a transparencia da janela -> qnto maior mais escuro. Varia
// de 0(100%) a 4(0%)
int est_geral = 0; // indica o estado geral da maquina e serve para controlar passos
// do motor

void main(void) //rotina principal
{
    config_lcd(); // chama funcao que envia comandos para o LCD reconhecer dados
// enviados

    retorna_motor(); /* chama funcao que levará motor ao nivel maximo de transparencia do
// sistema para evitar desconfiguracao em caso de desligamento de energia
// */

    linha1(); // chama funcao que enviara comando para que uma string seja escrita na
// linha 1 do LCD
    escreve_char("1 PARA ESCURECER");
    linha2(); // chama funcao que enviara comando para que uma string seja escrita
// na linha 2 do LCD
    escreve_char(" ");

    while(1) // loop principal do sistema
    {
```

```

if( bot1 == 1 )                //comando para escurecer a janela se botao 1 for acionado
{
    est_geral++;                /* incrementa 1 no na contagem de qual passo o motor parou
                                para que comece a rodar do proximo */

    if(est_geral <5)            // verifica se motor não atingiu o nivel maximo
    {
        if (transp >= 4)        // retorna valor da transparencia para 4 caso passe
            transp = 4 ;

        else                    // escurece janela
        {
            transp++;
            linha1();
            escreve_char(" ESCURECENDO P/ ");
            linha2();
            mostra_transp();
            incrementa_led(status);
            tempo();

            if (est_geral ==4)    // janela com nivel minimo de transparencia somente uma opção
            {
                linha1();
                escreve_char("2 PARA CLAREAR ");
                linha2();
                escreve_char("      ");
            }
            else                  // janela em nivel intermediario mostra duas opções
                retorna_tela_inicial();
        }
    }

    else                        // janela com nivel maximo de transparencia apresenta aviso
    {
        linha1();
        escreve_char("JANELA ESTA COM ");
        linha2();
        escreve_char(" 0% DE TRANSP. ");
        est_geral=4;
        tempo_mensag_limite();
        linha1();
        escreve_char("2 PARA CLAREAR ");
        linha2();
        escreve_char("      ");
    }
}

if( bot2 == 1 )                //comando para clarear a janela se botao 2 for acionado
{
    est_geral--;                /* incrementa 1 no na contagem de qual passo o motor parou
                                para que comece a rodar do proximo */

    if(est_geral >=0)
    {
        if (transp <= 0)        // retorna valor da transparencia para 0 caso passe
            transp = 0 ;

        else                    // clareia janela
        {
            transp--;
            linha1();
            escreve_char("CLAREANDO P/ ");
            linha2();
        }
    }
}

```

```

    mostra_transp();
    decrementa_led(status);
    tempo();
    if (est_geral == 0)                // janela com nivel minimo de transparencia somente uma opção
    {
        linha1();
        escreve_char("1 PARA ESCURECER");
        linha2();
        escreve_char("      ");
    }
    else                               // janela em nivel intermediario mostra duas opções
        retorna_tela_inicial();
}

else                                  // janela com nivel minimo de transparencia
{
    linha1();
    escreve_char("JANELA ESTA COM ");
    linha2();
    escreve_char(" 100% DE TRANSP.");
    est_geral = 0;
    tempo_mensag_limite();
    linha1();
    escreve_char("1 PARA ESCURECER");
    linha2();
    escreve_char("      ");
}
}
}
}

void escreve_char(char msn[])          // função para escrita no LCD
{
    unsigned char conta;                // utiliza char com 8 bits contando de 0 a 255
    for(conta=0; conta<16; conta++)
    {
        dado = msn[conta];
        wr_char();                      // chama função de escrita no LCD
    }
}

void mostra_transp(void)               // função para indicar para qual nivel de transparencia está indo a
janela
{
    switch(transp)
    {
        case 0:
            escreve_char("100% TRANSP.  ");
            break;

        case 1:
            escreve_char("75% TRANSP.  ");
            break;

        case 2:
            escreve_char("50% TRANSP.  ");
            break;
    }
}

```

```

case 3:
    escreve_char("25% TRANSP.  ");
    break;
case 4:
    escreve_char("0% TRANSP.  ");
    break;
}
}

void linha1(void)                // define o local de inicio de impressao na linha 1 do LCD
{
    dado = 0x80;
    wr_cmd();
}

void linha2(void)                // define o local de inicio de impressao na linha 2 do LCD
{
    dado = 0xc0;
    wr_cmd();
}

void config_lcd(void)            // função de comandos para configuração do Display
{

    dado = 0x38;
    wr_cmd();
    dado = 0x07;
    wr_cmd();
    dado = 0x0E;
    wr_cmd();
    dado = 0x06;
    wr_cmd();
    dado = 0x01;
    wr_cmd();

}

void delay_6us(void)             //rotina de delay de 2 microsegundos
{
    _asm                          //inicio de trecho assembly
        0$: mov  r0, #6
        1$: djnz r0, 1$
    _endasm;                      //fim do trecho assembly

    // 1 ciclo com 12Mhz = 1us => 1 ciclo com 27Mhz = 0.444444444444 microsegundos
    // #ciclos = 1c+(2c)*6
    // #ciclos = 13
    // #tempo = 13 * 0.444444444444 = 5.777 microsegundos
}

void delay_6ms(void)             //rotina de delay de 2 milisegundos
{
    _asm                          //inicio de trecho assembly
        1$: mov  r1, #27           //1c
        2$: mov  r2, #255         //1c
        3$: djnz r2, 3$           //2c

```

```

        djnz r1, 2$           //2c
_endasm;                     //fim do trecho assembly

// 1 ciclo com 12Mhz = 1us => 1 ciclo com 27Mhz = 0.444444444444 microsegundos
// #ciclos = 1c+[1c+(2c)*255+2c]*27
// #ciclos = 13852
// tempo = 13852 * 0.444444444444 = 6.156 milisegundos
}

void wr_char()               // função escreve string no Display
{
    rs = 1;
    en = 1;
    delay_6us();
    en = 0;
    delay_6ms();
}

void wr_cmd(void)            // função envia comando ao Display
{
    rs = 0;
    en = 1;
    delay_6us();
    en = 0;
    delay_6ms();
}

void tempo(void)             // Função para controle de tempo entre um passo e outro
{ int d;
    for(d=0;d<4;d++)
    {
        delay_6ms();
    }
}

void tempo_mensag_limite(void) // função de tempo para mostrar mensagem de limite de transparencia
{ int d;
    for(d=0;d<166;d++)        // aproximadamente 1 segundo
    {
        delay_6ms();
    }
}

void retorna_tela_inicial(void) // função mostra tela com as duas opções disponível
{
    linha1();
    escreve_char("1 PARA ESCURECER");
    linha2();
    escreve_char("2 PARA CLAREAR ");
}

void retorna_motor(void)     // função Inicializa Sistema
{
    linha1();
    escreve_char(" INICIALIZANDO ");
}

```



```

linha2();
escreve_char("SISTEMA AGUARDE!");
while(bot4 == 0 )
{
    int i;
    for(i=4;i>0;i--)
    {
        if(bot4 == 1)
        {
            status = i;
            break;
        }
        tempo();
        switch(i)
        {
            case 1:
                led1=1;
                led2=0;
                led3=0;
                led4=0;
                break;
            case 2:
                led1=0;
                led2=1;
                led3=0;
                led4=0;
                break;
            case 3:
                led1=0;
                led2=0;
                led3=1;
                led4=0;
                break;
            case 4:
                led1=0;
                led2=0;
                led3=0;
                led4=1;
                break;
            default:
                break;
        }
    }
}

void incrementa_led(int x)    // função escurece janela que retorna X (ultimo passo)
{
    int i,a,w=15;
    while(w>0)
    {
        for(i=x;i<5;i++)    // gira para desativar sensor
        {
            tempo();
            switch(i)
            {

```

```

    case 1:
        led1=1;
        led2=0;
        led3=0;
        led4=0;
        break;
    case 2:
        led1=0;
        led2=1;
        led3=0;
        led4=0;
        break;
    case 3:
        led1=0;
        led2=0;
        led3=1;
        led4=0;
        break;
    case 4:
        led1=0;
        led2=0;
        led3=0;
        led4=1;
        break;

    default:
        break;
}
x=1;
}
w--;
}

while(bot3 == 0) // continua girando janela até sensor ser acionado
{
    for(i=x;i<5;i++)
    {
        if (bot3 == 1)
        {
            status = i;
            break;
        }

        tempo();
        switch(i)
        {
            case 1:
                led1=1;
                led2=0;
                led3=0;
                led4=0;
                break;
            case 2:
                led1=0;
                led2=1;

```



```

        led3=0;
        led4=1;
        break;
    default:
        break;
    }
    x=4;
}
w--;
}
while(bot3 == 0)    // continua girando janela até sensor ser acionado
{
    for(i=x;i>0;i--)
    {
        if(bot3 == 1)
        {
            status = i;
            break;
        }
        tempo();
        switch(i)
        {
            case 1:
                led1=1;
                led2=0;
                led3=0;
                led4=0;
                break;
            case 2:
                led1=0;
                led2=1;
                led3=0;
                led4=0;
                break;
            case 3:
                led1=0;
                led2=0;
                led3=1;
                led4=0;
                break;
            case 4:
                led1=0;
                led2=0;
                led3=0;
                led4=1;
                break;
            default:
                break;
        }
    }
    x=4; // seta o valor para o final
}
}

```

ANEXO I – DATASHEET DO CI L298N

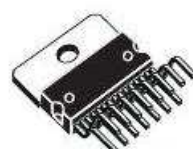

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-

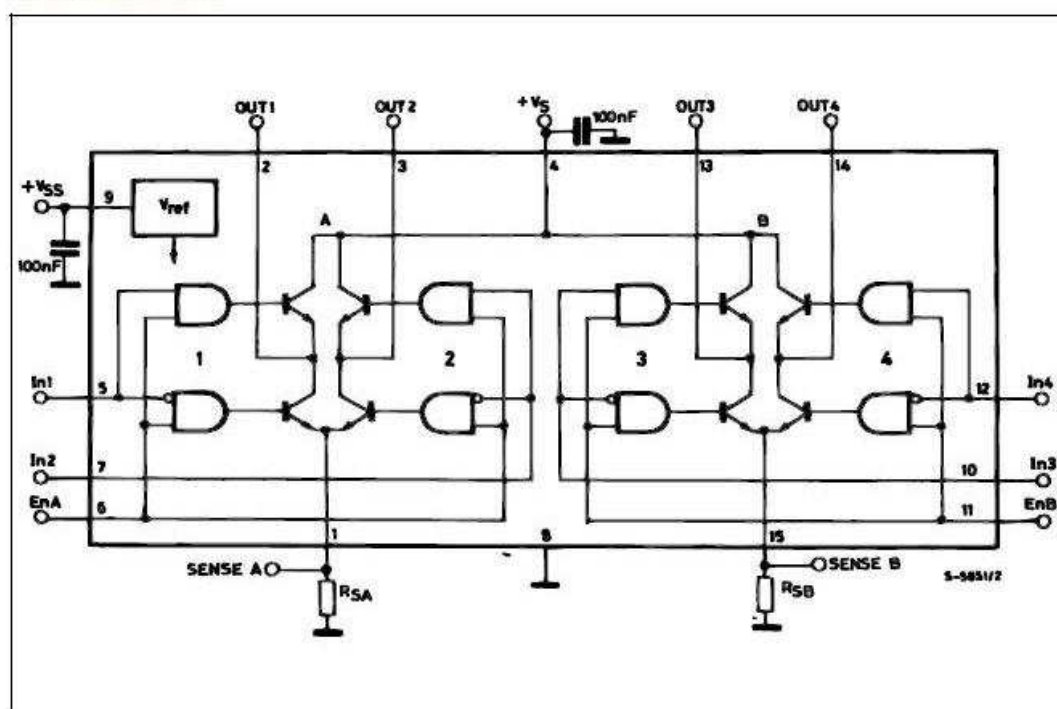

Multiwatt15

PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

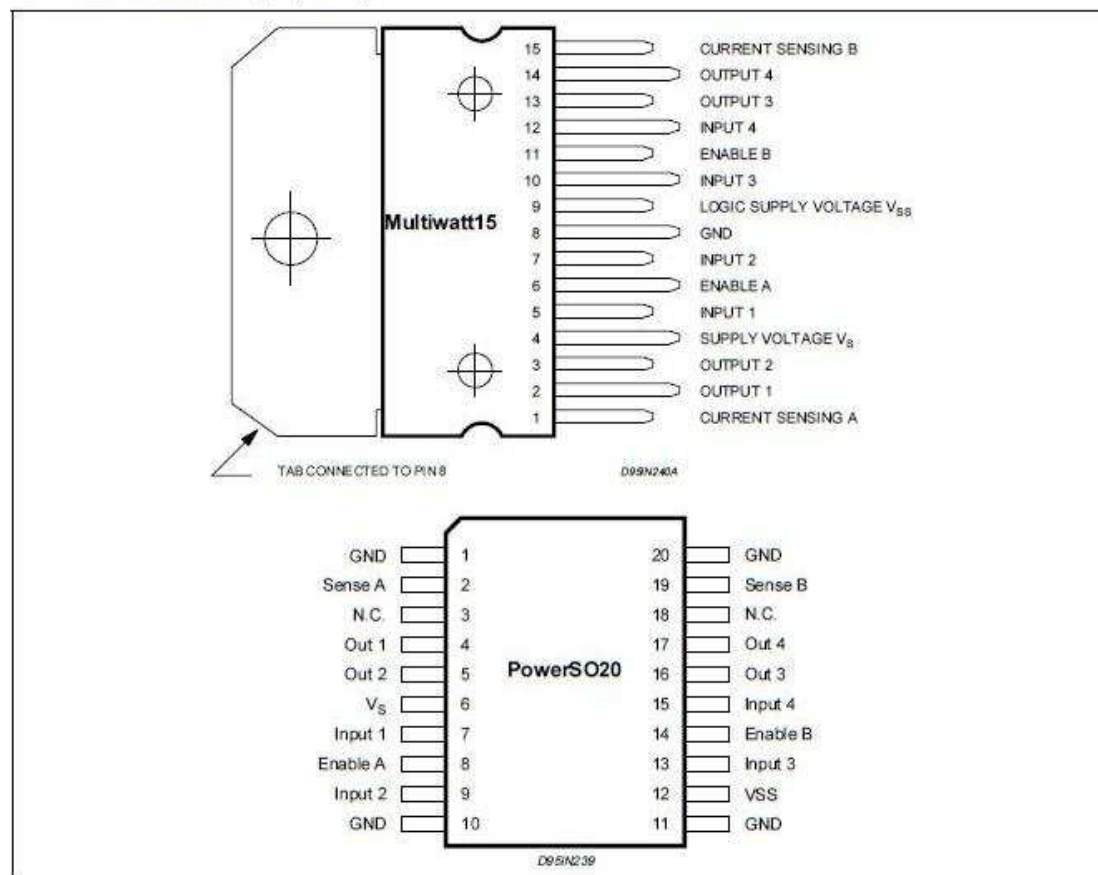
nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



L298**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel)		
	– Non Repetitive ($t = 100\mu s$)	3	A
	– Repetitive (80% on –20% off; $t_{on} = 10ms$)	2.5	A
	– DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)**THERMAL DATA**

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{\theta j-case}$	Thermal Resistance Junction-case	Max.	–	3	$^\circ C/W$
$R_{\theta j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate

L298

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
—	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_J = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _i = L V _i = H		13 50	22 70	mA mA
		V _{en} = L V _i = X			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0 V _i = L V _i = H		24 7	36 12	mA mA
		V _{en} = L V _i = X			6	mA
V _{IL}	Input Low Voltage (pins 5, 7, 10, 12)		−0.3		1.5	V
V _{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			−10	μA
I _{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} −0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		−0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			−10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} −0.6V		30	100	μA
V _{CEsat} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		−1 (1)		2	V

L298

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$T_1 (V_i)$	Source Current Turn-off Delay	$0.5 V_i$ to $0.9 I_L$ (2); (4)		1.5		μs
$T_2 (V_i)$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		0.2		μs
$T_3 (V_i)$	Source Current Turn-on Delay	$0.5 V_i$ to $0.1 I_L$ (2); (4)		2		μs
$T_4 (V_i)$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.7		μs
$T_5 (V_i)$	Sink Current Turn-off Delay	$0.5 V_i$ to $0.9 I_L$ (3); (4)		0.7		μs
$T_6 (V_i)$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.25		μs
$T_7 (V_i)$	Sink Current Turn-on Delay	$0.5 V_i$ to $0.9 I_L$ (3); (4)		1.6		μs
$T_8 (V_i)$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.2		μs
$f_c (V_i)$	Commutation Frequency	$I_L = 2A$		25	40	KHz
$T_1 (V_{en})$	Source Current Turn-off Delay	$0.5 V_{en}$ to $0.9 I_L$ (2); (4)		3		μs
$T_2 (V_{en})$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		1		μs
$T_3 (V_{en})$	Source Current Turn-on Delay	$0.5 V_{en}$ to $0.1 I_L$ (2); (4)		0.3		μs
$T_4 (V_{en})$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.4		μs
$T_5 (V_{en})$	Sink Current Turn-off Delay	$0.5 V_{en}$ to $0.9 I_L$ (3); (4)		2.2		μs
$T_6 (V_{en})$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.35		μs
$T_7 (V_{en})$	Sink Current Turn-on Delay	$0.5 V_{en}$ to $0.9 I_L$ (3); (4)		0.25		μs
$T_8 (V_{en})$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.1		μs

1) Sensing voltage can be $-1 V$ for $t \leq 50 \mu s$; in steady state $V_{sens} \min \geq -0.5 V$.

2) See fig. 2.

3) See fig. 4.

4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current

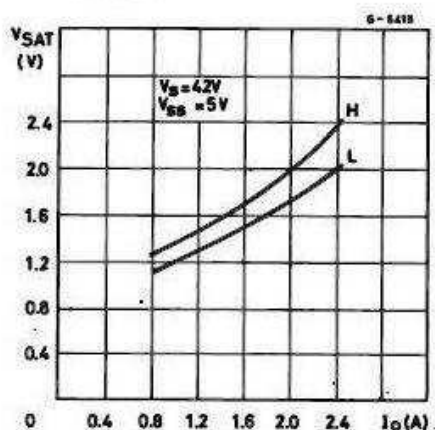
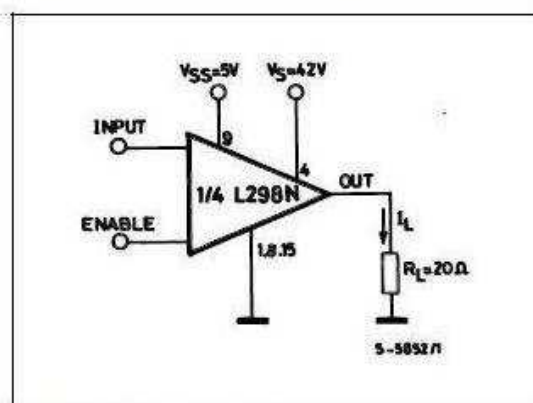


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

ANEXO II – DATASHEET DO MICROCONTROLADOR AT89S52

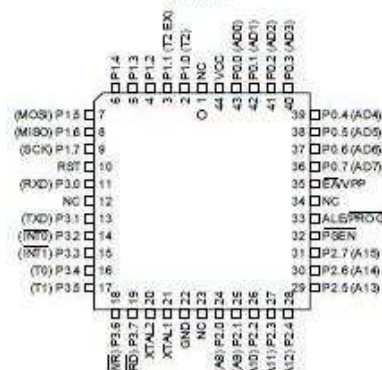


Pin Configurations

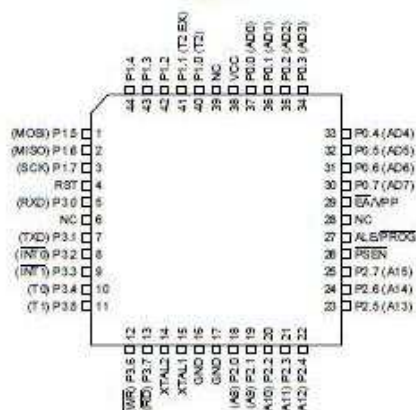
PDIP



PLCC

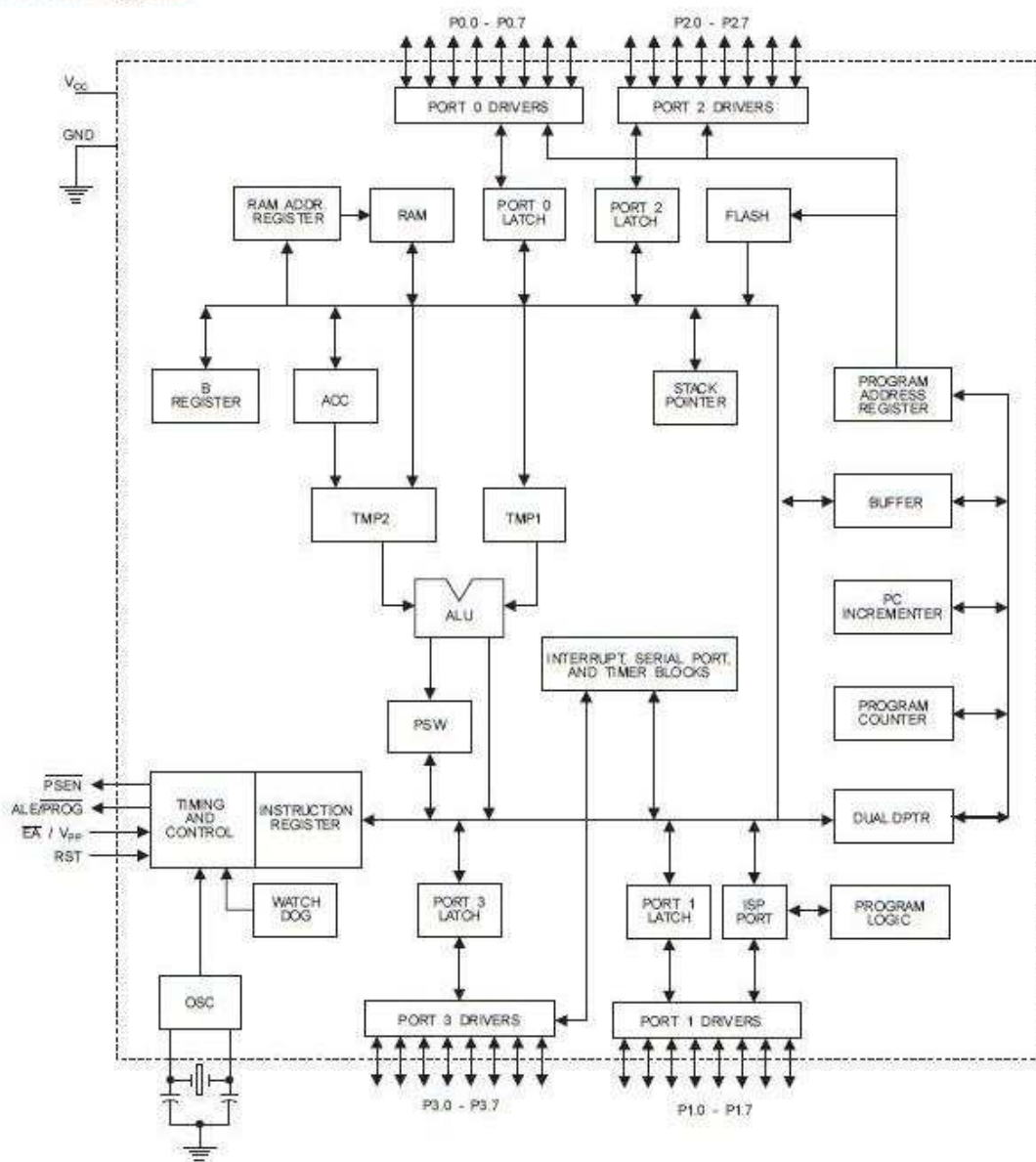


TQFP



AT89S52

Block Diagram





Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to

external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ R1), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/PROG

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is

AT89S52

weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable ($\overline{\text{PSEN}}$) is the read strobe to external program memory.

When the AT89S52 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA/VPP}}$

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH.

Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDRST XXXXXXXX	0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	PCON 0XXX0000	87H





Memory Organization

MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory

If the \overline{EA} pin is connected to GND, all program fetches are directed to external memory.

On the AT89S52, if \overline{EA} is connected to V_{CC} , program fetches to addresses 0000H through 1FFFFH are directed to internal memory and fetches to addresses 2000H through FFFFFH are to external memory.

Data Memory

The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions which use direct addressing access of the SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

AT89S52

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

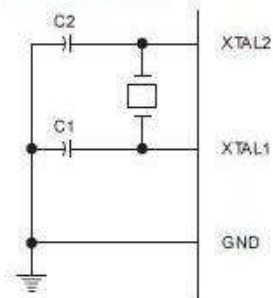
Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held

active long enough to allow the oscillator to restart and stabilize.

Figure 11. Oscillator Connections



Note: C1, C2 = 30 pF \pm 10 pF for Crystals
= 40 pF \pm 10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration

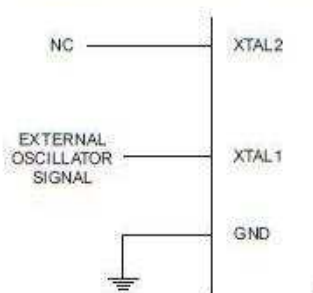


Table 6. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

ANEXO III – DATASHEET DO *DISPLAY* 16X2

SPECIFICATIONS FOR LIQUID CRYSTAL DISPLAY

PART NUMBER:

MGD1602B

DATE:

MAR. 17,2004

MGD1602B SERIES LCD MODULE

1.0 MECHANICAL SPECS

1. Overall Module Size	84.0mm(W) x 44.0mm(H) x max 13.5mm(D) for LED backlight version 84.0mm(W) x 44.0mm(H) x max 9.5mm(D) for reflective version
2. Dot Size	0.56mm(W) x 0.61mm(H)
3. Dot Pitch	0.61mm(W) x 0.66mm(H)
4. Duty	1/16
5. Controller IC	KS0066U or Equivalent
6. LC Fluid Options	TN, STN
7. Polarizer Options	Reflective, Transflective, Transmissive
8. Backlight Options	LED
9. Temperature Range Options	Standard temp.: (Operating: 0°C ~ 50°C; Storage: -20°C ~ 70°C) Wide temp.: (Operating: -20°C ~ 70°C; Storage: -30°C ~ 80°C)

2.0 ABSOLUTE MAXIMUM RATINGS

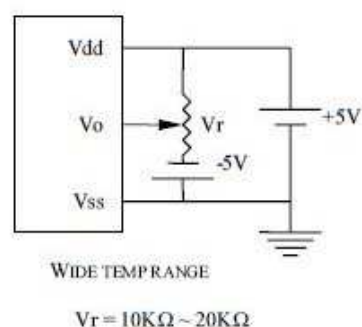
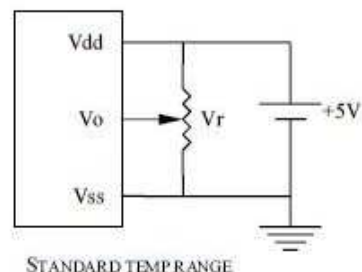
Item	Symbol	Min	Typ	Max	Unit
Operating temperature (Standard)	Top	0	-	50	°C
Storage temperature (Standard)	Tst	-20	-	70	°C
Operating temperature (Wide temperature)	Top	-20	-	70	°C
Storage temperature (Wide temperature)	Tst	-30	-	80	°C
Input voltage	Vin	Vss	-	Vdd	V
Supply voltage for logic	Vdd- Vss	2.7	-	5.5	V
Supply voltage for LCD drive	Vdd- Vo	3.0	4.5	6.5	V

MGD1602B SERIES LCD MODULE

7.0 PIN ASSIGNMENT

Pin No.	Symbol	Function
1	Vss	Ground
2	Vdd	+5V
3	Vo	LCD contrast adjust
4	RS	Register select
5	R/W	Read / write
6	E	Enable
7	DB0	Data bit 0
8	DB1	Data bit 1
9	DB2	Data bit 2
10	DB3	Data bit 3
11	DB4	Data bit 4
12	DB5	Data bit 5
13	DB6	Data bit 6
14	DB7	Data bit 7
+	BL+	Power Supply for BL+
-	BL-	Power Supply for BL-

8.0 POWER SUPPLY



9.0 TIMING CHARACTERISTICS

Item	Symbol	Test Condition	Min.	Typ.	Max.	Unit
Enable cycle time	t_c	Fig. a, Fig. b	500	-	-	ns
Enable pulse width	t_w	Fig. a, Fig. b	220	-	-	ns
Enable rise/fall time	t_r, t_f	Fig. a, Fig. b	-	-	25	ns
RS, R/W set up time	t_{su}	Fig. a, Fig. b	40	-	-	ns
RS, R/W hold time	t_{H}	Fig. a, Fig. b	10	-	-	ns
Data delay time	t_d	Fig. b	-	-	120	ns
Data set up time	t_{osu}	Fig. a	60	-	-	ns
Data hold time	t_{OH}	Fig. a, Fig. b	20	-	-	ns

